

Table des matières

Résumé.....	i
Abstract.....	ii
Remerciements.....	iii
Table des matières	v
Liste des tableaux.....	ix
Liste des figures	xiii
Liste des acronymes.....	xvi
Introduction.....	18
1. Introduction théorique.....	21
1.1 Notation et concepts.....	21
1.2 L'idée	22
1.3 Arbre de décision classique	23
1.3.1 Principe général	23
1.3.2 Critères de segmentation.....	24
1.3.3 L'élagage de l'arbre	26
1.4 Technique de boosting.....	28
1.4.1 ADABOOST	28
1.4.2 ADTree	29

1.5	Forêt aléatoire	29
1.6	Évaluation du modèle	30
1.6.1	Matrice de confusion	30
1.6.2	Courbe ROC	32
1.7	Sélection de variables	33
1.7.1	Filtre	35
1.7.2	Wrapper	36
1.7.3	Algorithme de recherche	37
2.	Réseau de test	40
2.1	EMTP-RV	40
2.2	Le réseau Anderson	40
2.3	Réseau Anderson Sous EMTP	41
2.3.1	Description du réseau modélisé	43
2.4	Simulations	50
2.4.1	Load-Flows	50
2.4.2	Temporel	54
2.4.3	Fichier DWJ	55
3.	Exploration des données	56
3.1	Importations des données sous MATLAB	56
3.2	Simulations non-terminées	58
3.3	Tri des simulations stables et instables	60
3.3.1	Introduction théorique au centre d'inertie	60
3.3.2	Critères d'instabilité	62
3.3.3	Application sur l'ensemble de données	64
3.3.4	Performances de discrimination	68
3.3.5	Vecteur Status	81
3.4	Ajout de nouvelles variables	81
3.4.1	Phi et Cosinus Phi	82
3.4.2	SCV et dSCV/dt	82

3.4.3	Fréquence inter-barres et fréquence filtrée	83
3.5	Création des différents ensembles	85
3.6	Exportation des données	87
4.	Modélisation et évaluation du prédicteur	88
4.1	KNIME	88
4.2	Baisse de la fréquence d'échantillonnage	90
4.2.1	C4.5 Pruned	90
4.2.2	C4.5 Unpruned	91
4.2.3	CART Pruned	92
4.2.4	CART Unpruned	93
4.2.5	ADABOOST	94
4.2.6	ADTree	95
4.2.7	10 forêts aléatoires	96
4.2.8	100 forêts aléatoires	97
4.2.9	Résumé et analyse	98
4.3	Réduction du nombre de variables	99
4.3.1	CFS - Best First	100
4.3.2	CFS - Rank Search	101
4.3.3	Sym - FCBF	102
4.3.4	Consistency - Best First	103
4.3.5	Consistency - Rank Search	104
4.3.6	Résumé du nombre de variables sélectionnées	105
4.3.7	Performances de prédiction	105
4.4	Validation	110
4.4.1	Générateur 1	111
4.4.2	Générateur 2	115
4.4.3	Générateur 3	118
4.4.4	Générateur 4	122
4.5	Réduction du taux de FP	125

Conclusion	i
Perspectives de recherches.....	iv
Bibliographie	v
Annexe 1	I
Annexe 2	IV
Annexe 3	VIII
Annexe 4	XIX
Annexe 5	XXI
Annexe 6	XXXV

Liste des tableaux

Table 1.1 Une matrice de confusion	31
Table 2.1 Paramètre des générateurs	43
Table 2.2 Paramètre des transformateurs.....	44
Table 2.3 Paramètres des lignes.....	45
Table 2.4 Charges Utilisés	47
Table 2.5 Mesure de base à chaque générateur	48
Table 2.6 Mesure des composantes symétriques et fréquence	48
Table 2.7 Paramètres de load-flow 1 ^{ère} simulation	51
Table 2.8 Résumé du load-flow EMTP de la 1 ^{ère} simulation	51
Table 2.9 Paramètres de load-flow 2 ^{ème} simulation.....	52
Table 2.10 Résumé du load-flow EMTP de la 2 ^{ème} simulation	52
Table 2.11 Paramètres de load-flow 3 ^{ème} simulation.....	53
Table 2.12 Résumé du load-flow EMTP de la 3 ^{ème}	53

Table 3.1 Simulations finies avant 20 secondes	59
Table 3.2 Répartition des simulations stables et instables	66
Table 3.3 Numéro de simulations des machines ayant divergé	66
Table 3.4 Tableau des simulations divergentes selon le COI pour le générateur 1	69
Table 3.5 Tableau des simulations divergentes selon le COI pour le générateur 1	70
Table 3.6 Tableau des simulations divergentes selon le COI pour le générateur 1	71
Table 3.7 Tableau des simulations divergentes selon le COI pour le générateur 2	72
Table 3.8 Tableau des simulations divergentes selon le COI pour le générateur 2	73
Table 3.9 Tableau des simulations divergentes selon le COI pour le générateur 2	74
Table 3.10 Tableau des simulations divergentes selon le COI pour le générateur 3	75
Table 3.11 Tableau des simulations divergentes selon le COI pour le générateur 3	76
Table 3.12 Tableau des simulations divergentes selon le COI pour le générateur 3	77
Table 3.13 Tableau des simulations divergentes selon le COI pour le générateur 4	78
Table 3.14 Tableau des simulations divergentes selon le COI pour le générateur 4	79
Table 3.15 Tableau des simulations divergentes selon le COI pour le générateur 4	80
Table 3.16 Variables prédictives et le numéro de colonne	85
Table 3.17 Répartition des simulations au total	86
Table 3.18 Nombre de simulations dédiées à l'apprentissage	86
Table 3.19 Nombre de simulations dédiées au test	86

Table 3.20 Nombre de simulations dédiées à la validation	86
Table 4.1 Mesures AUC en fonction de la fréquence C4.5 Pruned	90
Table 4.2 Mesures AUC en fonction de la fréquence C4.5 Unpruned	91
Table 4.3 Mesures AUC en fonction de la fréquence CART Pruned	92
Table 4.4 Mesures AUC en fonction de la fréquence CART Unpruned	93
Table 4.5 Mesures AUC en fonction de la fréquence ADABOOST	94
Table 4.6 Mesures AUC en fonction de la fréquence ADTree	95
Table 4.7 Mesures AUC en fonction de la fréquence 10 forêts aléatoires	96
Table 4.8 Mesures AUC en fonction de la fréquence 100 forêts aléatoires	97
Table 4.9 Mesure AUC pour le générateur 1	98
Table 4.10 Variables sélectionnées CFS Best First	100
Table 4.11 Variables sélectionnées CFS Rank Search	102
Table 4.12 Variables sélectionnées Consistency Best First	103
Table 4.13 Variables sélectionnées Consistency Rank Search	104
Table 4.14 Nombre de variables sélectionnées par filtrage	105
Table 4.15 AUC des 4 générateurs avec filtrage de variable et forêts aléatoires	106
Table 4.16 AUC des 4 générateurs avec filtrage de variable et ADtree	106
Table 4.17 Résumé des variables sélectionnées pour chaque générateur	109
Table 4.18 Matrice de confusion pour le générateur 1 sur l'ensemble de validations	113

Table 4.19 Taux mesurés par la matrice de confusions du générateur 1	114
Table 4.20 Matrice de confusion pour le générateur 2 sur l'ensemble de validation	116
Table 4.21 Taux mesurés par la matrice de confusion du générateur 2.....	116
Table 4.22 Matrice de confusion pour le générateur 3 sur l'ensemble de validation	120
Table 4.23 Taux mesurés par la matrice de confusions du générateur 3	120
Table 4.24 Matrice de confusion pour le générateur 4 sur l'ensemble de validaiton	124
Table 4.25 Taux mesurés par la matrice de confusions du générateur 4	124
Table 4.26 Taux de 100 RF pour une base de données Standard et Equilibrée.....	126
Table 4.27 Taux de 10 RF pour une base de données Standard et Equilibrée.....	126
Table 4.28 Taux de 2 RF pour une base de données Standard et Equilibrée.....	127
Table 4.29 Taux pour ADABOOST pour une base de données Standard et Equilibrée	127
Table 4.30 Taux pour ADTree pour une base de données Standard et Equilibrée	128

Liste des figures

Figure 1.1 Une partie d'un arbre de décision.....	24
Figure 1.2 Sur-apprentissage de l'arbre de décision	27
Figure 1.3 Compromis entre complexité et taux d'erreur	28
Figure 1.4 Courbe ROC	32
Figure 1.5 Sélection de variables sous WEKA [12]	34
Figure 1.6 Méthode de sélection linéaire [17]	38
Figure 2.1 Schéma unifilaire du réseau Anderson.....	41
Figure 2.2 Réseau Anderson modélisé sous EMTP-RV	42
Figure 2.3 Sous-circuit de ligne.....	45
Figure 2.4 Mesure des fréquences et des composantes symétriques de V et I	49
Figure 2.5 Mesure de P, Q, I et V en RMS et temporel.....	50
Figure 3.1 Workspace MATLAB après importation	57
Figure 3.2 Séparation de X en VectElect et VectMec	58

Figure 3.3 Caractéristique des vitesses. En bleu les simulations arrêtées avant 20 sec.....	60
Figure 3.4 COIA des simulations pour $t < 20[s]$	63
Figure 3.5 COI Ω des simulations pour $t < 20[s]$	63
Figure 3.6 Agrandissement de COI Ω pour trouver le seuil	64
Figure 3.7 COIA de toutes les simulations	65
Figure 3.8 COI Ω de toutes les simulations.....	65
Figure 3.9 Vitesse des 180 générateurs : Stable Rouge, Instable en Bleu	68
Figure 3.10 Vecteur status et court-circuit.....	81
Figure 3.11 Illustration du SCV.....	83
Figure 3.12 Espace disque utilisé par les fichiers exportés	87
Figure 4.1 Workflow en utilisant KNIME.....	89
Figure 4.2 Courbes ROC C4.5 Pruned	90
Figure 4.3 Courbes ROC C4.5 Unpruned.....	91
Figure 4.4 Courbes ROC CART Pruned	92
Figure 4.5 Courbes ROC CART Unpruned.....	93
Figure 4.6 Courbes ROC ADABOOST	94
Figure 4.7 Courbes ROC ADTree	95
Figure 4.8 Courbes ROC 10 forêts aléatoires	96
Figure 4.9 Courbes ROC 100 forêts aléatoires	97

Figure 4.10 AUC en fonction du $\log_2$ (fréquence).....	98
Figure 4.11 Comparaison de performances des réducteurs de variables - Forêts aléatoires	107
Figure 4.12 Comparaison de performances des réducteurs de variables - ADTree	107
Figure 4.13 Work flow KNIME pour l'ensemble de validation.....	110
Figure 4.14 Status et Winner pour le générateur 1	111
Figure 4.15 Exemple de fonctionnement du RPS pour la simulation 1151 - Générateur 1	112
Figure 4.16 Exemple de fonctionnement du RPS pour la simulation 1060 - Générateur 1	113
Figure 4.17 Courbe ROC du générateur 1 sur l'ensemble de validation	114
Figure 4.18 Status et Winner pour le générateur 2	115
Figure 4.19 Exemple de fonctionnement du RPS pour la simulation 1052 - Générateur 2	116
Figure 4.20 Courbe ROC du générateur 2 sur l'ensemble de validation	117
Figure 4.21 Status et Winner pour le générateur 3	118
Figure 4.22 Exemple de fonctionnement du RPS pour la simulation 1013 - Générateur 3	119
Figure 4.23 Exemple de fonctionnement du RPS pour la simulation 1117 - Générateur 3	120
Figure 4.24 Courbe ROC du générateur 3 sur l'ensemble de validation	121
Figure 4.25 Status et Winner pour le générateur 4	122
Figure 4.26 Exemple de fonctionnement du RPS pour la simulation 1040 - Générateur 4	122
Figure 4.27 Exemple de fonctionnement du RPS pour la simulation 1099 - Générateur 4	123
Figure 4.28 Courbe ROC du générateur 4 sur l'ensemble de validation	124

Liste des acronymes

0,d,q	Composante symétrique : Homopolaire, Directe et Inverse
ADABOOST	Adaptive Boosting
ADTree	Alternating Decision Tree
AUC	Area Under the Curve
B	Nombre d'arbres créés pour les forêts aléatoires
BF	Best First
C	Variable à prédire
CART	Classification And Regression Tree
CFS	Correlation-based Feature Subset Selection
COIA	Center of Inertia of Angle
COI Ω	Center of Inertia of Speed
Consistency	Consistency Subset Eval
cy	Cycles
FA	Forêts Aléatoires
FCBF	Fast Correlation Based Filter
FN	Faux Négatifs
FP	Faux Positifs
Gen	Générateur
H	Entropie
I	Courant simple
IG	Information Gain
k	Nombre de classes de la variable à prédire
M	Nombre de variables sélectionnées pour les forêts aléatoires
n	Nombre d'individus de l'ensemble

p	Nombre de variables prédictives de l'ensemble
P	Puissance active d'une phase
P	Prunned
PV	Barre de type Puissance Tension
Q	Puissance réactive d'une phase
r	Coefficient de Pearson
RMS	Root Mean Square
ROC	Receiver Operating Characteristic
RS	Rank Search
SCV	Swing Center Voltage
Symm	Symmetrical Uncert Attribute Set Eval
T	Temporel
UP	Unprunned
V	Tension simple
VN	Vrai-Négatif
VP	Vrai-Positif
V δ	Barre de type Tension Angle
X	Variables prédictives de l'ensemble
$\delta_{COI} - \omega_{COI}$	Angle - Vitesse moyenne de toutes les machines
$\delta_j - \omega_j$	Angle - Vitesse moyenne des machines de la même région
$\delta_j^{COI} - \omega_j^{COI}$	Centre d'inertie d'angle - vitesse de la machine j
φ	Angle courant-tension
Ω	Individus de l'ensemble

Introduction

Nous vivons dans un monde de données. Une quantité inimaginable de gigabites est sauvegardée chaque jour, de la caisse du centre d'achat, jusqu'à l'historique de notre navigateur, en passant par les statistiques sportives, le domaine de la finance, de la sociologie et des télécommunications. Tous les domaines sauvegardent chaque donnée car les éléments de stockage sont bon marché et il est possible, avec des outils statistiques et les ressources informatiques actuelles, de faire parler ces grandes bases de données et par conséquent, celles-ci ont une grande valeur.

En effet, les méthodes de data mining existaient depuis longtemps. C'est leur application sur les grandes bases de données qui posait problème [1]. Parmi de nombreuses recherches dans le domaine de l'intelligence artificielle, on peut citer que, dans les années 50, McCulloch et Pits furent les premiers à publier des travaux sur les réseaux de neurones, puis Sonquist et Morgan firent de même avec les arbres de décisions dans les années 60. En 1984, Breiman [2] publiait les arbres de décision pouvant faire de la classification et de la régression, et ce même chercheur développa en 2001 une amélioration des arbres de décision, soit les forêts aléatoires [3]. Avec les disques durs, les processeurs et les mémoires vives actuelles, il est possible d'utiliser pleinement ces outils de travail.

Le réseau électrique, quant à lui, n'a pas beaucoup évolué. Le système (source, lignes électriques, transformateurs, et récepteur) est toujours utilisé. Cependant, la demande en électricité croît sans cesse et les lignes ainsi que les centrales de production arrivent à saturation car, pour des raisons écologiques et environnementales, il n'est pas possible d'en construire de nouvelles.

En ayant peu de réserve de puissance, nous nous exposons à des problèmes d'instabilité. En effet, après une perturbation telle qu'une ouverture de ligne, la perte d'une charge, ou encore un court-circuit, les machines électriques freinent ou accélèrent [4]. Si une trop brusque perturbation survient, les machines qui ne sont pas assez fortes se font entraîner par le réseau et décrochent, menant ainsi à de très grosses coupures de courant comme celle survenue à New York en 2003 privant ainsi d'électricité 55 millions de consommateurs.

Le relais que nous tentons de développer peut aider à prévoir une telle instabilité, nous permettant ainsi de déconnecter certaines parties du réseau pour le décharger si les machines ont tendance à décélérer ou couper des postes de production si elles ont tendance à accélérer et ainsi, limiter et réduire le nombre d'utilisateurs affectés.

Plusieurs relais de protection de synchronisme ont déjà été développés. Ceux-ci mesurent les caractéristiques concentriques et le taux de changement de l'impédance ou encore $Rdot$, pour ne citer qu'elle [5].

En créant nous même une base de données de contingences, il est possible d'entraîner un modèle prédictif afin de détecter les cas stables et instables. Dans le document [6], l'auteur crée une base de données de load-flow et, en fonction de la durée et de l'état de charge du réseau, il est possible de prédire les courts-circuits résultant en une instabilité.

Avec ce document, nous tenterons de confirmer la faisabilité d'un relais de protection de synchronisme en temps réel avec pour avantages une facilité de mise en œuvre, une détection plus rapide des instabilités et donc, la prise de décision plus rapide concernant les charges ou les machines à déconnecter.

Ce rapport sera présenté de la manière suivante : le premier chapitre décrit les notations qui sont utilisées tout au long du rapport. Il y décrit aussi l'idée selon laquelle le relais de protection de synchronisme fonctionnera et une introduction théorique à la fouille de données et aux arbres de décisions est présentée au lecteur.

Le chapitre 2 expliquera le système qui sera utilisé et les divers éléments qui le composent (les lignes, les transformateurs, les générateurs etc.) seront détaillés. De plus, ce chapitre présentera les diverses simulations temporelles et de load-flow qui ont été effectuées.

Dans le chapitre 3, le lecteur trouvera les informations décrivant le processus de création, de manipulation, de transformation et d'agrandissement de la base de données. Ce chapitre présentera en plus la méthode de discrimination des simulations stables et instables.

Le chapitre 4 présentera la modélisation des prédicteurs en utilisant les bases de données créées, présentera l'application des diverses techniques de réduction de variables ainsi que les performances sur le système.

Finalement une conclusion synthétisera les travaux et les résultats présentés dans ce document. D'autres pistes de recherche ou de continuité dans cette filière y seront exposées.

1. Introduction théorique

Ce premier chapitre a pour but d'introduire les notions théoriques nécessaires au lecteur. La première section présente les notations ainsi que la nomenclature qui sera utilisée tout au long de ce rapport.

Nous présenterons ensuite l'idée et le principe selon lesquels le relais de protection de synchronisme sera mis en œuvre.

Finalement, les dernières sections présentent sommairement les techniques de modélisation utilisées tel que les arbres de décisions classiques, les méthodes de boosting et bagging, les forêts aléatoires, les techniques servant à évaluer les modèles, ainsi que la réduction du nombre de variable.

1.1 Notation et concepts

Nous entendons par ensemble la matrice de donnée composée de variables et d'individus. Dans notre cas, nous avons l'ensemble d'apprentissage qui sera utilisé pour former le modèle prédictif, l'ensemble de test qui servira à évaluer les performances de celui-ci et l'ensemble de validation qui va nous servir à visualiser les classifications.

Les colonnes des ensembles sont appelées des variables prédictives alors que les lignes sont des individus. Cette matrice est de taille $n \times p$.

$X = \text{Variables prédictives}$ $C = \text{Variable à prédire}$

		x_1	x_2	x_3	$\dots$	x_p		C
$\Omega = \text{Individus}$	ω_1							k_1
	ω_2							k_2
	ω_3							k_3
	$\dots$							$\dots$
	ω_n							k_n

La colonne C à prédire peut être composée de plusieurs classes k . Dans notre cas, la variable à prédire est composée de deux classes, soit stable et positive définie comme 0, soit instable et négative définie comme 1.

La prédiction de classification est le terme utilisé lorsque nous faisons appel au modèle d'apprentissage pour prévoir la classe à la sortie du modèle pour de nouveaux individus.

1.2 L'idée

En utilisant la méthode de centre d'inertie [7], les simulations effectuées sont au préalable manuellement classées stables et instables.

Pour les simulations stables, la variable à prédire de chaque individu et pour chaque générateur est définie à 0 alors que pour les cas instables celle-ci est définie à 1. Les individus instables avant la fin du court-circuit sont ignorés. Le modèle de prédiction est ensuite entraîné à classer les cas individus après individus.

Appliqué en temps réel, après un court-circuit, chaque nouvel individu échantillonné est classé par le modèle de prédiction dans un état stable ou instable jusqu'à ce qu'une décision d'ouverture des disjoncteurs soit prise.

1.3 Arbre de décision classique

Un arbre de décision est une méthode simple et populaire dans le domaine du data mining. Les avantages d'un arbre de décision sont les suivants [8] :

- Il peut être utilisé pour la classification, la régression et le clustering ;
- Il fournit des règles explicites de classement facilitant sa lisibilité ;
- Il supporte les données hétérogènes ;
- Il supporte les données manquantes et les effets non linéaires ainsi que les variables non-normalisées ;
- Il est capable de gérer les grandes quantités de données ;
- Il nécessite peu de ressources informatiques.

Les inconvénients majeurs sont les suivants :

- Problème de stabilité sur les petites bases de données ;
- Le noeud du niveau $k+1$ dépend du niveau k .
- La variance des données peut grandement influencer la qualité de prédiction

1.3.1 Principe général

Un arbre de décision est une approche haut en bas où la variable prédictive ayant le plus grand pouvoir de segmentation des individus est testée dans le premier noeud appelé noeud racine. Ce noeud génère n classes dans lesquelles la deuxième variable avec le plus grand pouvoir de segmentation est testé, formant à son tour n classes. Cette opération est répétée jusqu'à ce que la segmentation des individus ne soit plus possible. Les terminaisons de l'arbre de décision dans lequel l'affectation de la prédiction est effectuée sont appelées noeuds terminaux ou feuilles.

A titre illustratif, la racine et quelques nœuds d'un arbre de décision sont présentés dans la figure ci-dessous.

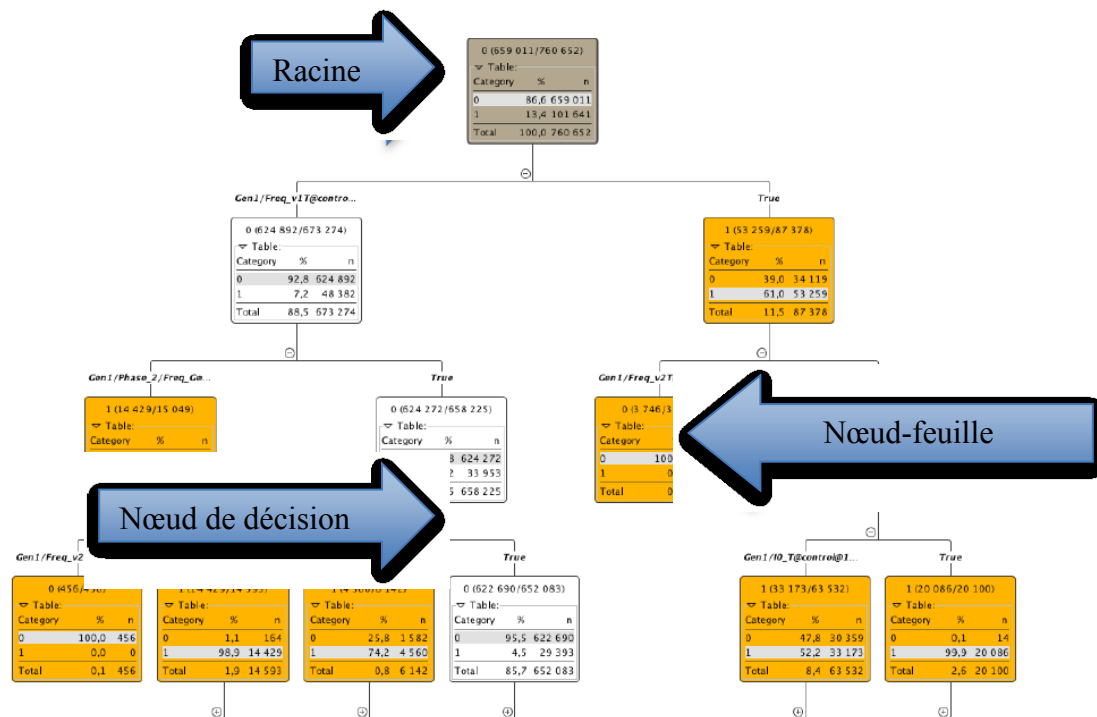


Figure 1.1 Une partie d'un arbre de décision

Par exemple, dans la figure ci-dessus, les cases en blanc présentent le chemin possible d'une décision.

1.3.2 Critères de segmentation

Afin de déterminer la meilleure discrimination des classes dans un noeud, plusieurs méthodes sont possibles:

1.3.2.1 Indice de Gini

Cet indice mesure la probabilité que deux individus choisis aléatoirement dans un noeud appartiennent à deux classes différentes. Plus l'indice de Gini est bas, plus le noeud est pur. Chaque noeud-fils doit donner la plus grande hausse de pureté, donc, la plus grande baisse de l'indice de Gini. L'indice de Gini est calculé de la manière suivante [9] :

$$Gini(Noeud) = 1 - \sum_i f_i^2 = \sum_{i \neq j} f_i f_j \quad (1.1)$$

Ou f_i , $i=1$ à p sont les fréquences relatives dans le noeud des p classes à prédire et pour avoir une séparation, il faut minimiser l'équation suivante:

$$Gini(séparation) = \sum_{i=1}^k \frac{n_k}{n} Gini(k^{ème} noeud) \quad (1.2)$$

Les arbres de décision de type CART (Classification And Regression Tree) inventés par Breiman [2] utilisent ce critère de séparation.

1.3.2.2 Entropie

Une autre méthode de discrimination consiste à remplacer l'indice de Gini par l'entropie et chercher à la minimiser dans les noeuds-fils. L'arbre de décision C4.5 et C5.0 utilisent ce critère de décision.

L'entropie pour une variable X est donnée par l'équation suivante :

$$H(X) = - \sum_i P(x_i) \cdot \log_2(P(x_i)) \quad (1.3)$$

L'entropie de X sachant Y est donnée par l'équation suivante :

$$H(X|Y) = - \sum_j P(y_j) \cdot \sum_i P(x_i|y_j) \cdot \log_2(P(x_i|y_j)) \quad (1.4)$$

1.3.2.3 Le gain d'information

La quantité d'information apportée par Y par rapport à X est appelée gain d'information et est définie comme suit :

$$IG(X|Y) = H(X) - H(X|Y) \quad (1.5)$$

Dans tous les cas, l'algorithme arrête de chercher un noeud-fils lorsque le critère de séparation est inférieur à un seuil défini par l'utilisateur ou lorsque tous les nœuds feuilles ne contiennent qu'une classe de la variable cible.

D'autres critères de séparation ont été développés et peuvent être utilisés. Parmi eux, nous pouvons citer [8]:

- Mesure de la distance ;
- Indice de Twoing ;
- Critère du χ^2 ;
- Critère orthogonal ;
- Critère de l'aire sous la courbe ROC.

1.3.3 L'élagage de l'arbre

Le sur-apprentissage consiste à mémoriser les liens entre la variable prédictive et la variable à prédire au lieu de comprendre et de généraliser. Dans le cas de notre arbre de décision, cela signifie que notre arbre de décision est trop grand et celui-ci épouse les fluctuations de l'échantillon d'apprentissage comme illustré dans la figure suivante.

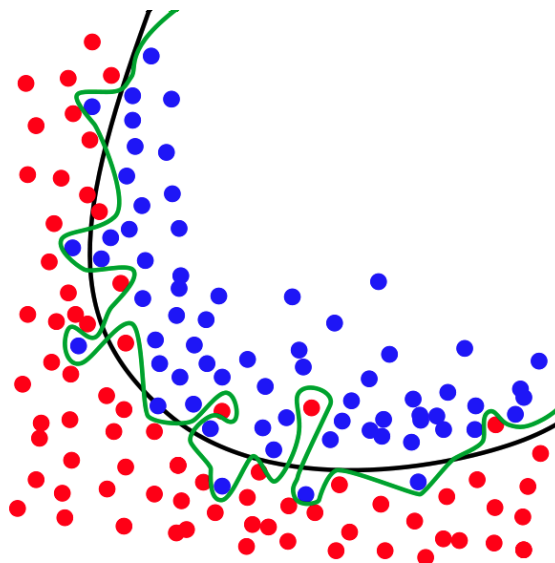


Figure 1.2 Sur-apprentissage de l'arbre de décision

La ligne verte correspond à un sur-apprentissage des données. Le taux d'erreur de classification est fiable, mais en contre partie, le modèle est complexe et il n'est plus capable de généraliser. La ligne noire est un bon compromis entre un modèle trop complexe et un modèle trop simple.

Pour un arbre de décision, le principe d'élagage détermine le juste compromis d'un modèle. Concrètement, cela signifie que les noeuds terminaux trop spécifiques sont regroupés et généralisés dans des noeuds plus hauts dans la hiérarchie de l'arbre.

Après avoir entraîné un arbre de décision et mesuré son taux d'erreur (courbe bleue) en fonction du nombre de branches de celui-ci, il faut ensuite le tester avec l'ensemble de test et déterminer l'erreur de test en fonction de la complexité. Comme illustré dans la figure suivante, la ligne pointillée représente le minima de l'erreur de test (courbe rouge) et par conséquent, la taille idéale de l'arbre.

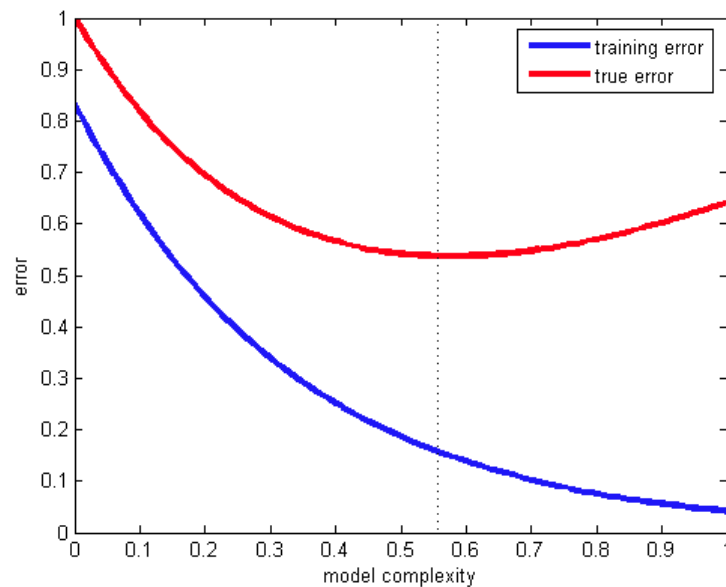


Figure 1.3 Compromis entre complexité et taux d'erreur

La courbe bleue est appelée courbe de restitution de l'erreur alors que la courbe rouge est appelée erreur de test.

1.4 Technique de boosting

1.4.1 ADABOOST

Cette méthode utilise des règles simples pour faire une approximation des fonctions complexes. En effet, son principe est d'agréger ensemble une multitude de faibles hypothèses prédictives faites sur l'ensemble de données afin d'en créer une plus forte [10].

Pour le cas des arbres de décision, l'algorithme ADABOOST construit une multitude de petits arbres de décision qui à eux seuls ont une capacité de prédiction faible, mais, mis ensemble, ceux-ci gagnent en performance.

Au départ, tout l'ensemble d'apprentissage se voit distribuer un poids égal. Après avoir trouvé une hypothèse forte, l'algorithme augmente le poids des exemples ne satisfaisant pas

à l'hypothèse précédente et celui-ci cherche une autre hypothèse plus faible répondant au mieux au reste de l'ensemble et ainsi de suite. La prédiction finale est faite en pondérant les résultats des petits arbres.

1.4.2 ADTree

Les arbres de décision alternatifs, ou ADTrees, offrent la lisibilité d'un arbre de décision classique jumelé aux bénéfices apportés par la méthode de boosting et contrairement aux arbres de type Adaboost, les ADTrees sont très rapides même sur une grande base de données. Leur principal désavantage est qu'ils ne supportent que des variables à prédire à deux classes.

À la différence d'un arbre classique, les noeuds de prédiction peuvent être suivis d'un autre nœud de décision en alternance, d'où le nom arbre de décision alternatif.

Après qu'un individu soit inséré dans le nœud racine, on lui comptabilise un score initial qui est le biais. Par la suite, une variable est testée et le score dans le nœud de prédiction est ajouté au score initial. S'il y a un autre nœud de décision après le nœud, l'algorithme teste la condition suivante et continue ainsi jusqu'au nœud feuille. Lorsque l'individu a atteint le nœud final, la prédiction de la classe est prise en comparant la somme des scores obtenus dans les nœuds de prédiction à +1 et à -1. Si le score est positif, telle décision est prise, sinon, c'est l'autre.

1.5 Forêt aléatoire

La méthode de random forest, ou forêt aléatoire, développée par Breiman [3] repose sur 3 principes soit : les arbres de décisions conventionnels, les méthodes de bagging ainsi qu'un sous-ensemble de variables prédictives aléatoires.

D'abord, il faut échantillonner, avec remplacement, un nombre d'individus égal à l'ensemble d'apprentissage, soit n . Cette méthode est appelée bagging, ou Bootstrap Aggregating. Par la suite, un arbre de décision est construit sur cet échantillon, dans lequel

un nombre M de variables aléatoires est utilisé dans chaque noeud de décision. Le nombre de variable M dépend du nombre total de variable d'entrée X_P et est calculé par la formule suivante :

$$M = \log_2(X_P + 1) \quad (1.6)$$

Cette opération est refaite afin de construire B arbres et la prédiction finale pour une entrée est la moyenne, dans le cas d'une régression, ou un vote dans le cas d'une classification parmi les B arbres.

Cette méthode offre d'excellentes performances car elle permet de maintenir une variance constante dans tout l'ensemble de données. Malheureusement, étant donné que B arbres de décision doivent être construits, cette méthode est très lente et très gourmande en ressources informatiques. De plus, cette méthode n'est pas intuitive comme un arbre de décision classique.

1.6 Évaluation du modèle

1.6.1 Matrice de confusion

La matrice de confusion est un outil performant et bien adapté aux problèmes de classification [11]. Les lignes de la matrice de confusion représentent les prédictions alors que les colonnes représentent les classes réelles, dans notre cas, une matrice binaire de 2x2. Nous considérons un individu instable (1) comme négatif et un individu stable comme positif (0).

Classe réelle \ Classe prédite	0	1	Total prédit
0	Vrai Positif VP	Faux Négatif FN	VP + FN
1	Faux Positif FP	Vrai Négatif VN	FP + VN
Total réel	VP + FP	FN+VN	n

Table 1.1 Une matrice de confusion

La matrice de confusion est obtenue grâce à l'ensemble de donnée de test. Celle-ci est passée dans le modèle prédictif et la variable à prédire est comparée à la variable obtenue par le modèle.

Par exemple, si le prédicteur donne 1 à la sortie pour un individu prédit alors que dans la donnée de test la classe de l'individu en question est 0, il s'agit d'un faux négatif. Grâce à la matrice de confusion, nous pouvons calculer les différents taux:

$$\text{Précision} = \frac{VP + VN}{n} \quad (1.7)$$

$$\text{Taux d'erreur} = \frac{FP + FN}{n} \quad (1.8)$$

$$\text{Sensibilité} = \frac{VP}{VP + FN} \quad (1.9)$$

$$\text{Spécificité} = \frac{VN}{VN + FP} \quad (1.10)$$

$$\text{Taux de faux positif} = \frac{FP}{FP + VN} \quad (1.11)$$

$$\text{Taux de faux négatif} = \frac{FN}{FN + VP} \quad (1.12)$$

La sensibilité d'un modèle nous indique la probabilité du modèle à prédire un vrai individu lorsque celui-ci est effectivement vrai. Son opposé s'appelle la spécificité. Le taux d'erreur, quant à lui, indique la quantité de faux positif par rapport à l'ensemble des individus. Le taux de vrai et de faux positifs nous indique la quantité d'individus qui sont prédits vrai alors qu'en réalité ils sont faux et vice-versa.

1.6.2 Courbe ROC

La courbe ROC où Receiver Operating Characteristic – Caractéristique de Fonctionnement du Récepteur est un outil inventé pendant la seconde guerre mondiale et servait à montrer la séparation d'un signal avec le bruit. Appliquée à notre cas, la courbe ROC nous indique l'indice de performance d'un modèle en comparant le taux de faux positif en abscisse et la sensibilité en ordonné.

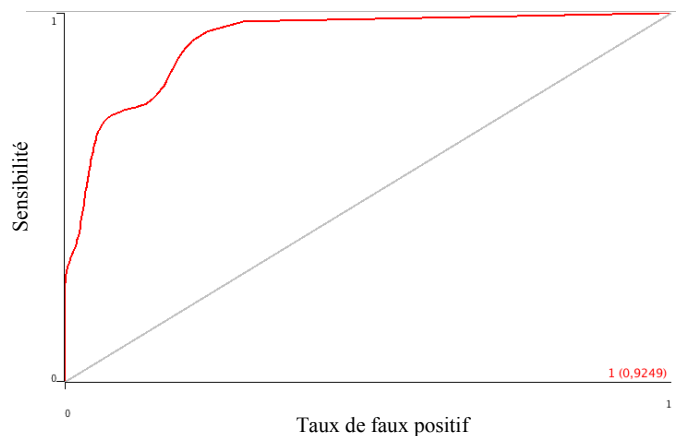


Figure 1.4 Courbe ROC

Cette courbe est obtenue en faisant varier le seuil de score. Pour chaque pas de score, une matrice de confusion est créée et le taux de faux positif et la sensibilité sont reportés sur le graphique [9]. Si pour chaque pas de seuil, le taux de faux positif est égal à la sensibilité et la réponse du modèle sera aléatoire. Ceci correspond à la diagonale du graphique précédent. Idéalement, pour un score très faible, il faut obtenir le maximum de sensibilité et un minimum de taux de faux positif, ce qui correspond à un point (0,1) sur l'image précédente.

Pour comparer différents modèles l'un à l'autre, la matrice de confusion nous donne plusieurs coefficients de performances à comparer, tel que par exemple le taux de faux-positif ou la spécificité. La courbe ROC, quant à elle, grâce à son aire sous la courbe AUC (Area Under the Curve), centralise toutes ces données en un seul nombre compris de 0 à 1.

- 0.5 - si le modèle a une réponse aléatoire ;
- 1 – s'il s'agit d'un modèle parfait.

1.7 Sélection de variables

Au cours du temps, plusieurs méthodes permettant de sélectionner les variables les plus pertinentes ont été développées comme le montre la figure 1.5. Cette figure est une liste non-exhaustive des algorithmes utilisables dans WEKA.

Cette technique permet de réduire la dimension de la matrice d'apprentissage, ce qui a pour avantage de :

- Réduire les ressources informatiques nécessaires (RAM, processeur, disque dur) ;
- Réduire le temps d'apprentissage du modèle ;
- Améliorer la lisibilité du modèle ;
- Améliorer la capacité de généralisation du modèle et donc, la robustesse ;
- Retirer les variables redondantes et non pertinentes.

La branche Single Attribute Evaluators s'occupe de lister les variables en fonction de leur pertinence et de la quantité d'informations qu'elles apportent au modèle une à une. Ces méthodes ne sont pas pertinentes pour notre utilisation et par conséquent, elles ne seront pas développées.

La branche Subset Attribute Evaluators va chercher plusieurs variables qui, utilisées ensemble, offrent la meilleure performance de prédiction.

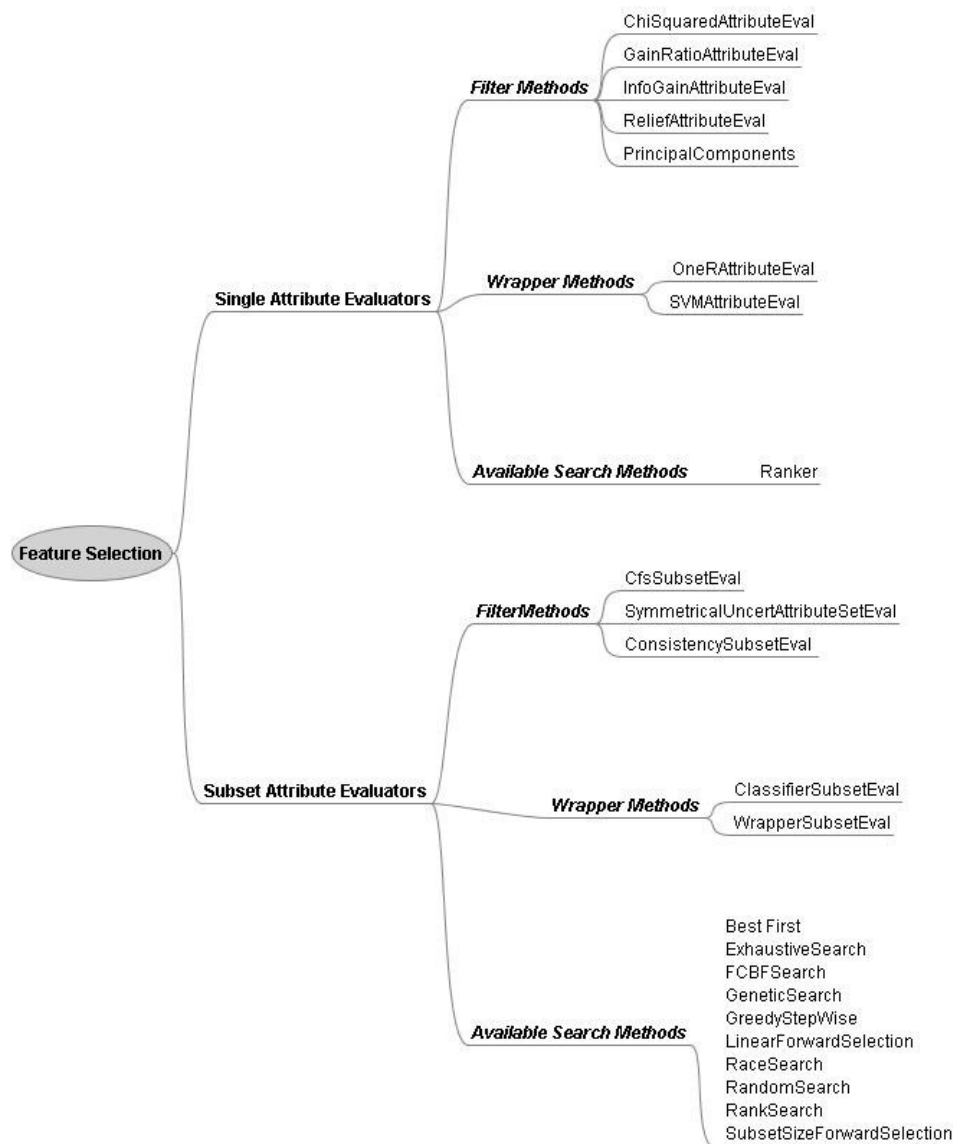


Figure 1.5 Sélection de variables sous WEKA [12]

1.7.1 Filtre

L'approche de type Filtre sélectionne au préalable le meilleur ensemble de variable indépendamment du prédicteur placé après. Les avantages de cette méthode sont sa constance ainsi que sa rapidité. De plus, selon les documents [13] et [14], la méthode Filtre n'est pas loin d'être aussi performante pour un coût en calcul bien moindre que la méthode Wrapper présentée dans le sous-chapitre 1.7.2. Pour la suite de ce projet, cette méthode filtre sera utilisée.

1.7.1.1 Cfs Subset Eval

La méthode Cfs Subset Eval qui veut dire Correlation-based Feature Subset Selection, recherche l'aptitude d'une variable prédictive à être hautement corrélée avec la variable à prédire mais faiblement corrélée entre les différentes variable prédictives [14]. Cette méthode utilise le coefficient de corrélation de Pearson r entre les deux variables X et Y calculé comme suit [15]:

$$r = \frac{\sum_i (x_i - \bar{x}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_i (x_i - \bar{x}_i)^2} \sqrt{\sum_i (y_i - \bar{y}_i)^2}} \quad (1.13)$$

Où $\bar{x}_i$ est la moyenne de X et $\bar{y}_i$ est la moyenne de Y et le coefficient r est contenu dans l'intervalle $[-1 ; 1]$. Si X et Y sont très corrélés, r vaudra 1 ou -1. Si par contre X est indépendant de Y , r vaudra 0. L'avantage de cette méthode est qu'elle permet d'enlever les variables qui contribuent peu au résultat et elle permet de réduire la redondance des variables. Son désavantage principal est qu'elle ne peut pas travailler avec des individus non-numériques.

1.7.1.2 Symmetrical Uncert Attribute Set Eval

Cette méthode utilise l'incertitude symétrique [15] de deux variable X et Y qui est mesurée par :

$$SU(X, Y) = 2 \cdot \left[\frac{IG(X|Y)}{H(X) + H(Y)} \right] \quad (1.14)$$

Sachant que IG est le gain d'information calculé par la formule (1.4) et H , l'entropie calculée par (1.3).

1.7.1.3 Consistency Subset Eval

Cette méthode évalue la qualité d'un sous-ensemble d'attributs en utilisant leur niveau de constance par rapport à la constance de tous l'ensemble d'apprentissage [16]. Si celui-ci est plus faible, cela signifie que, pour une plus faible quantité de variables, les mêmes résultats sont obtenus.

1.7.2 Wrapper

La méthode Wrapper sélectionne récursivement le meilleur ensemble de variable pour optimiser la réponse du prédicteur placé après en vérifiant par exemple, l'erreur, la précision ou l'aire sous la courbe. Par conséquent, pour chaque type de prédicteur, un sous-ensemble différent doit être choisi.

Cette méthode est intéressante quand la base de données est assez petite car elle permet de gagner en précision pour un temps de calcul relativement faible. Cependant, étant donné la très grande taille de notre base de données, ce processus est beaucoup trop long pour le gain final et par conséquent, les algorithmes de la méthode Wrapper ne seront pas détaillés.

1.7.3 Algorithme de recherche

Que ce soit pour la méthode Filtre ou Wrapper, un algorithme de recherche de l'ensemble de variable doit être utilisé. Les sous-chapitres suivants expliquent brièvement le fonctionnement de ces différentes méthodes. Les méthodes Exhaustive Search et Random Search étant beaucoup trop gourmandes en ressources ainsi que les méthodes Race Search et Genetic Search qui elles, sont plutôt adaptées pour les méthodes Wrapper, ne seront pas développées.

1.7.3.1 Greedy Stepwise

L'idée principale de cette méthode de recherche est de prendre la variable la plus prometteuse et d'y ajouter récursivement une autre variable qui améliore le plus le modèle jusqu'à ce que l'ajout de nouvelles variables n'apporte pas plus de information pertinente. Il est aussi possible de prendre tous les jeux de données et de retirer à chaque fois la variable la plus pertinente.

1.7.3.2 Best First

L'idée de base de l'algorithme Best First est la même que du Greedy Stepwise. Cependant, il offre la possibilité d'aller en arrière et d'explorer un autre chemin d'amélioration de prédiction ce qui lui permet d'éviter d'être bloqué dans un minima local.

1.7.3.3 Linear Forward Selection

L'algorithme utilise une méthode filtre pour classer les N variables en fonction de leur pertinence. Par la suite, seules les k meilleures variables sont utilisées comme entrées à l'algorithme de sélection linéaire et à chaque boucle, la variable la plus intéressante est retirée. Il est aussi possible de garder k constant en ajoutant à chaque fois une variable restante de $N-K$.

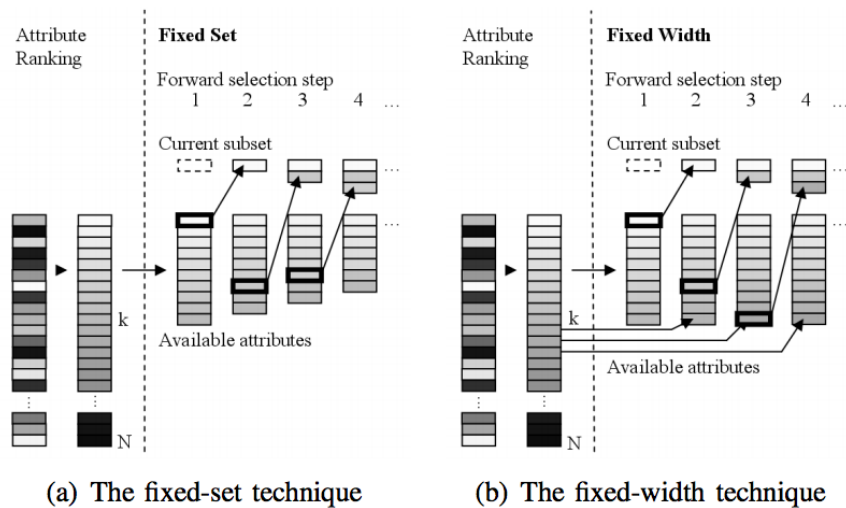


Figure 1.6 Méthode de sélection linéaire [17]

Cette méthode offre l'avantage de filtrer au préalable un grand nombre de variables et par conséquent de limiter le nombre de boucles dans l'algorithme de recherche.

Une extension à cet algorithme s'appelle Subset Size Forward Selection. Il effectue à l'intérieur de l'algorithme une validation croisée et, dans chaque échantillon, l'algorithme effectue une Linear Forward Selection afin de déterminer le meilleur sous-ensemble. Finalement, une Linear Forward Selection est effectuée dans tout l'ensemble de données afin de déterminer la taille du sous-ensemble optimal.

1.7.3.4 FCBF Search

Cette méthode appelée Fast Correlation Based Filter classe les variables par ordre de pertinence par rapport à la variable cible en utilisant l'incertitude symétrique. Par la suite, ces variables les plus pertinentes sont vérifiées à nouveau afin d'enlever toutes les variables redondantes.

1.7.3.5 Rank Search

Cet algorithme retourne toutes les variables classées par ordre de pertinence sans tenir compte de leur corrélation entre variable ou entre la variable cible. Par la suite, l'algorithme évalue la qualité du sous-ensemble de variable et ajoute la deuxième variable la plus pertinente et ainsi de suite.

Après essai sur KNIME, les combinaisons d'un algorithme de recherche et d'une méthode de classification suivantes peuvent être utilisées :

- FCBF fonctionne seulement avec la méthode Symmetrical Uncert Attribute Set Eval ;
- Best First et Rank search peuvent être utilisés avec Consistency Subset Eval et CFSSubseteval.

Au total, nous avons cinq manières différentes de filtrer les données. Les résultats de ces méthodes seront comparés plus tard dans ce document.

2. Réseau de test

La première section de ce chapitre présente EMTP-RV, le programme qui sera utilisé afin d'effectuer les simulations. Par après, le réseau Anderson [18] ainsi que la modélisation de tous les éléments EMTP-RV sera présenté.

Finalement, le dernier sous-chapitre présente les Loads flow ainsi que les simulations qui sont effectuées et qui serviront de base de données afin d'entraîner le modèle de prédiction.

2.1 EMTP-RV

Le programme utilisé pour effectuer les différentes contingences est EMTP-RV (ElectroMagnetic Transient Program – Restructured Version). Il a été développé par Hydro-Québec en 2003 et distribué par le DCG (Developement Coordination Groupe). Il a été choisi car c'est un programme qui offre de grandes performances en termes de rapidité de simulation et de grande précision pour de grands réseaux électriques [19]. De plus, il offre la possibilité de résoudre des problèmes de load-flows ainsi que des problèmes dynamiques.

2.2 Le réseau Anderson

Le réseau Anderson [18] caractérise le réseau nord-américain du fait de ces très grands générateurs reliés entre eux par de très longues lignes. Du fait de la grande impédance des lignes, ce réseau est sujet à de nombreuses oscillations de fréquences entre les générateurs. C'est la raison principale pour laquelle il a été choisi comme réseau de test.

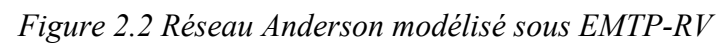


Figure 2.2 Réseau Anderson modélisé sous EMTP-RV

2.3.1 Description du réseau modélisé

2.3.1.1 Générateurs

La zone 1 est composée de deux générateurs de 1300 [MVA] et de 4400 [MVA] reliés entre eux par 1200 [km] de lignes. Le générateur de la zone 2 est relié aux générateurs de la zone 1 au travers d'une ligne de 500 [km] et le générateur de la zone 3 est quant à lui relié au générateur 2 par une ligne de 600 [km]. Les générateurs de la zone 2 et de la zone 3 sont reliés entre eux par une ligne de 1600 [km].

La table suivante montre les caractéristiques des différents générateurs.

	Générateur 1	Générateur 4	Générateur 2	Générateur 3
Puissance [MVA]	60 000	70 000	1 300	4 400
Nb de pôles [-]	54		64	
$R_a / X_l / X_0$ [pu]	0.03 / 0.149 / 0.113		0.011 / 0.14 / 0.127	
$X_d / X_d' / X_d''$ [pu]	1.028 / 0.34 / 0.253		1.1 / 0.323 / 0.229	
$X_q / X_q' / X_q''$ [pu]	0.544 / 0.5 / 0.298		0.416 / 0.4 / 0.212	
T_{d0}' / T_{d0}'' [s]	7.5 / 0.07		5.6 / 0.09	
T_{q0}' / T_{q0}'' [s]	7.5 / 0.09		5.6 / 0.09	
Constante H [pu]	4.1		3.2	

Table 2.1 Paramètre des générateurs

Ces paramètres correspondent à un test à circuit ouvert. Toutes les machines sont connectées en étoile avec mise à la terre de leur point milieu et la tension aux bornes de toutes les machines est de 13.8 [kV] RMS ligne-ligne.

2.3.1.2 Transformateurs

Les transformateurs ont un couplage Triangle/Étoile avec neutre mis à la terre. Chaque transformateur a une puissance nominale égale à la puissance du générateur en amont.

La tension au primaire est de 13.8 [kV] RMS ligne-ligne et la tension au secondaire est de 500 [kV] RMS ligne-ligne. Les caractéristiques des différents transformateurs sont présentées ci-dessous.

	Transfo. 1	Transfo. 2	Transfo. 3	Transfo. 4
Puissance [MVA]	60 000	1 300	4 400	70 000
Résistance au primaire [pu]	0.001	0.001	0.001	0.001
Résistance au secondaire [pu]	0.001	0.001	0.001	0.001
Réactance au primaire [pu]	0.001	0.001	0.001	0.001
Réactance au secondaire [pu]	0.11	0.082	0.082	0.09

Table 2.2 Paramètre des transformateurs

2.3.1.3 Lignes

Les lignes utilisées dans le modèle du réseau électrique ont une tension nominale de 500 [kV] et sont longues de 500 [km] pour la ligne C et 600 [km] pour les lignes A, B, D, E, F et G. Ces lignes sont de type à paramètres distribués (CP-Line). Ce type de ligne permet une grande vitesse de simulation avec peu d'erreur de calcul lorsque la fréquence varie peu. Les lignes sont transposées et les paramètres utilisés pour définir la ligne sont la résistance, l'inductance et la capacité linéique dans leur composante direct et homopolaire. Le tableau ci-dessous montre les paramètres utilisés :

	Longueur [km]	R' [Ω]	L' [H]	C' [F]
Homopolaire	300	0.35936	0.00266216	3.37889e-06
Direct	300	0.01613	0.0009218	1.23e-08

Table 2.3 Paramètres des lignes

Un sous-circuit est créé comme modèle comme illustré dans la figure suivante. Il contient :

- 2 lignes de type CP ;
- 1 compensateur série modélisé par une capacité triphasée ;
- 4 interrupteurs triphasés en série ;
- 2 interrupteurs triphasés de mise à terre avec en série élément RLC ;
- 3 interrupteurs monophasés en série.

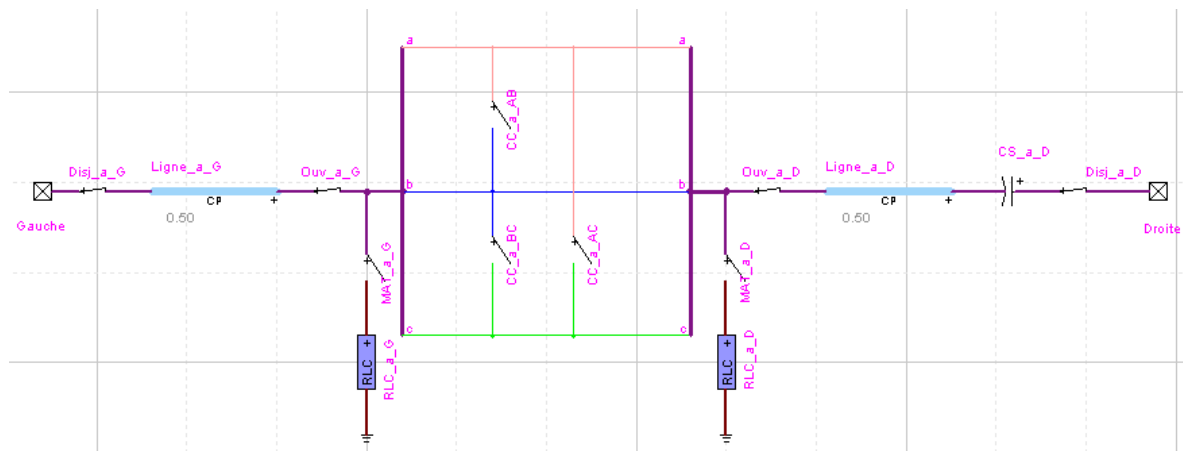


Figure 2.3 Sous-circuit de ligne

Ce sous-circuit est sauvegardé et réutilisé pour toutes les lignes. Il permet, en contrôlant les différents interrupteurs, d'obtenir un grand nombre de courts-circuits tel que par exemple une ouverture de ligne, un court-circuit à la terre franc ou au travers d'une impédance ou un court-circuit inter-phases. De plus, en spécifiant la longueur de la ligne à gauche et à droite, il est possible de contrôler la distance à laquelle un court-circuit survient.

Les désavantages consistent en un plus grand nombre de lignes à calculer pour le simulateur et par conséquent, des simulations plus longues et un pas de simulation est limité au minimum à un temps égal à la constante de temps de la ligne. Cependant, ces deux inconvénients sont largement compensés par la modularité qu'offre ce sous-système.

Dans notre cas, la fréquence choisie est de 3840 [Hz], ce qui correspond à un temps d'échantillonnage de 260.4167 [μ s].

À tout cela, nous ajoutons une compensation série de 30% avec condensateurs à chaque ligne avec une capacité 0.1824 [F] pour les lignes à 600 [km] de long et 0.152 [F] pour les lignes à 250 [km] de long.

2.3.1.4 SVC

Le SVC (Static VAR Compensator) ou Compensateur Statique d'énergie réactive est un dispositif utilisé pour injecter ou retirer de l'énergie réactive d'une barre d'un réseau de grande puissance. Cela nous permet d'avoir un contrôle de tension à la barre à laquelle il est connecté. Dans notre cas, nous imposons une tension de référence de 1.02 p.u et le compensateur statique s'occupe, grâce à son régulateur interne, d'injecter jusqu'à 1000 [kVAR] ou d'absorber jusqu'à 500 [kVAR] afin de maintenir la référence. Il est aussi important de noter que la tension à laquelle le SVC est pleinement capacitif est de 0.95 p.u. En outre, les gains proportionnels et intégrateurs sont de 12 et 2800. Les autres paramètres sont laissés par défaut.

2.3.1.5 Charges

Les charges utilisées pour la simulation sont de type PQ dans lesquelles nous imposons la tension nominale, soit 500 [kV] RMS ligne-ligne ainsi que la puissance active et réactive consommée par celle-ci. Le tableau suivant résume les différentes charges :

Nom	P [MW]	Q [MVAR]	Zone	Barre connectée
Load 1	40 000	8 000	2	Barre 1
Load 2	500	100	1	Barre 2
Load 3	2 750	550	1	Barre 3
Load 4	1 750	200	1	Barre 4
Load 5	50 000	12 000	3	Barre 5

Table 2.4 Charges utilisés

La puissance totale de la zone 2 est de 5000 [MW] et 850 [MVAR]. La puissance totale consommée de tout le réseau est de 95 [GW] et 20.85 [MVAR] soit 97.25 [GVA]. La puissance apparente totale produite par les machines est de 135.7 [MVA].

2.3.1.6 Sous-circuit de mesures

Le sous-système présenté dans la Figure 2.4 est mis en série après le secondaire de chaque transformateur. Il permet de mesurer les paramètres suivants :

- Courants et tensions simples RMS et temporels (12 mesures) ;
- Puissances actives et réactives simples RMS et temporelles (12 mesures) ;
- Composantes symétriques des courants et tensions RMS et temporels (12 mesures) ;
- Fréquences de chaque phase (3 mesures).

Trente-neuf mesures sont prises pour chaque générateur soit un total de 156 mesures.

Phase_1/I_RMS	Phase_2/I_RMS	Phase_3/I_RMS
Phase_1/V_RMS	Phase_2/V_RMS	Phase_3/V_RMS
Phase_1/P_RMS	Phase_2/P_RMS	Phase_3/P_RMS
Phase_1/Q_RMS	Phase_2/Q_RMS	Phase_3/Q_RMS
Phase_1/I_T	Phase_2/I_T	Phase_3/I_T
Phase_1/V_T	Phase_2/V_T	Phase_3/V_T
Phase_1/P_T	Phase_2/P_T	Phase_3/P_T
Phase_1/Q_T	Phase_2/Q_T	Phase_3/Q_T

Table 2.5 Mesure de base à chaque générateur

Les variables suivantes sont mesurées à l'aide des variables précédentes :

I0_RMS	V0_RMS
Id_RMS	Vd_RMS
Iq_RMS	Vq_RMS
I0_T	V0_T
Id_T	Vd_T
Iq_T	Vq_T
Freq_v1T	
Freq_v2T	
Freq_v3T	

Table 2.6 Mesure des composantes symétriques et fréquence

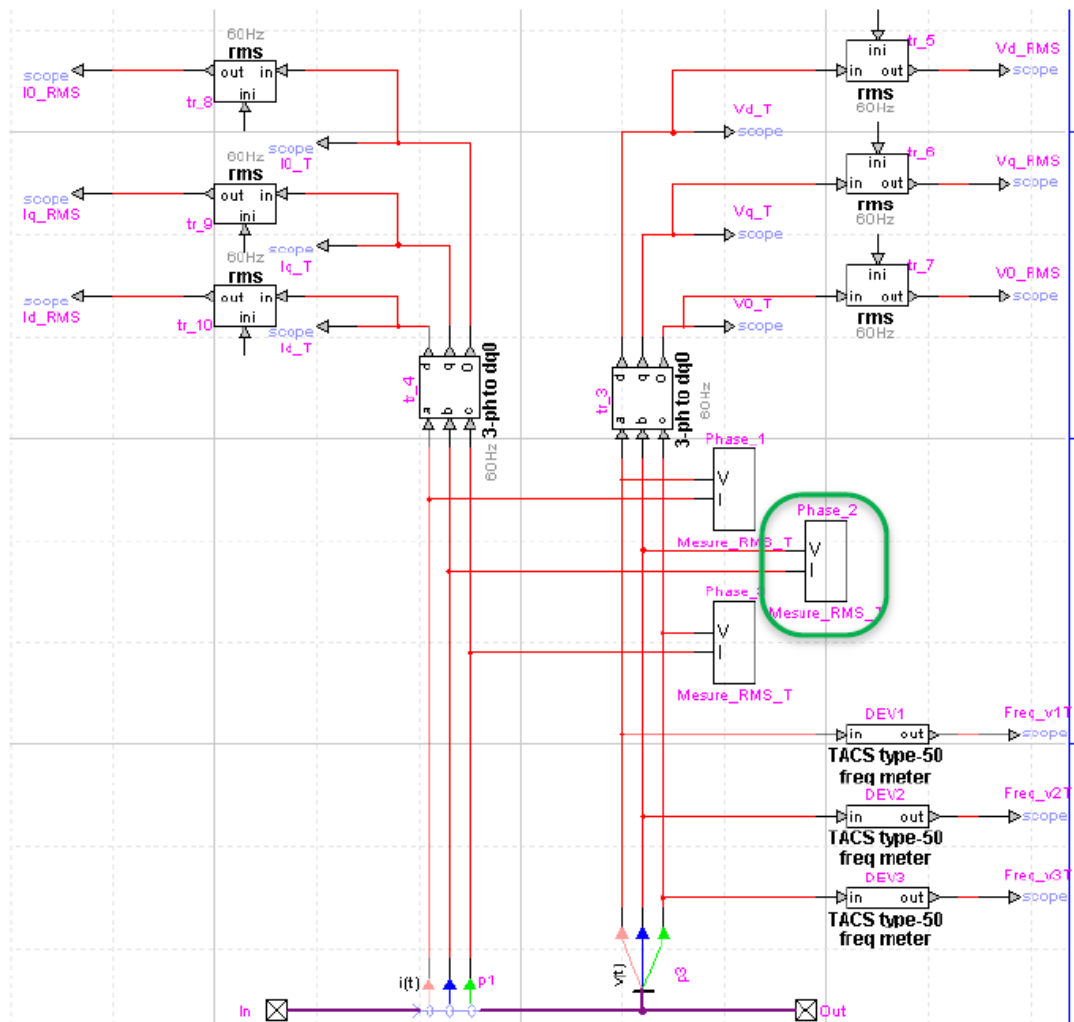


Figure 2.4 Mesure des fréquences et des composantes symétriques de V et I

La Figure 2.5 représente le sous-système encadré en vert dans la Figure 2.4

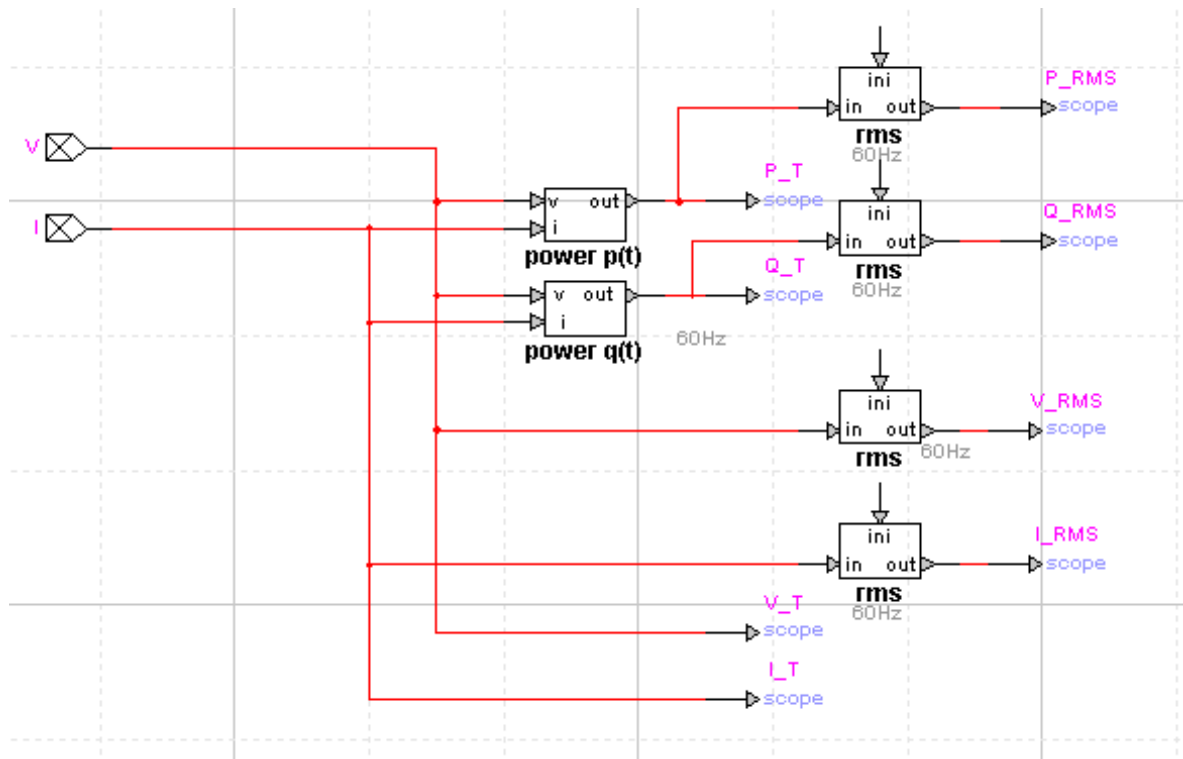


Figure 2.5 Mesure de P , Q , I et V en RMS et temporel

2.4 Simulations

2.4.1 Load-Flows

Afin d'obtenir la base de données, 3 loads-flows différents ont été simulés. Quand le bus de référence est placé aux bornes de la machine 1, les autres barres de type PV sont réglées de sorte à ce que les machines fournissent le maximum de puissance active. Cette mesure est prise afin que le réseau fonctionne toujours à la limite de l'instabilité et par conséquent lorsqu'un court-circuit survient, les machines auront tendance à décrocher plus facilement que si elles ont de la puissance en réserve.

2.4.1.1 Référence en zone 1

Le bus Slack ou $V\delta$ est placé aux bornes de la machine 4 et les autres bus sont de type PV. Les valeurs suivantes sont indiquées en RMS ligne-ligne et la puissance est triphasée :

Type	Générateur 1	Générateur 2	Générateur 3	Générateur 4
Type	PV	PV	$V\delta$	PV
Paramètres	13.8 [kV]	13.8 [kV]	13.8 [kV]	13.8 [kV]
	40 000 [MW]	1 000 [MW]	0 [°]	52 000 [MW]

Table 2.7 Paramètres de load-flow 1^{ère} simulation

Le tableau ci-dessus est le résultat obtenu après simulation du load-flow :

Device	Type	Vabc (kVRMSLL,deg)		P (W)	Q (VAR)	Eabc (kVRMSLL,deg)		Iabc (A,deg)	
LF_gen3	Slack	1.38E+01	-3.35E-13	2.59E+09	9.18E+08	1.71E+01	2.11E+01	1.63E+05	-1.95E+01
		1.38E+01	-1.20E+02			1.71E+01	-9.89E+01	1.63E+05	-1.40E+02
		1.38E+01	1.20E+02			1.71E+01	1.41E+02	1.63E+05	1.00E+02
LF1_svc	PVbus	5.10E+02	-2.72E+01	6.49E+00	7.58E+08	5.10E+02	-2.72E+01	1.21E+03	-1.17E+02
		5.10E+02	-1.47E+02			5.10E+02	-1.47E+02	1.21E+03	1.23E+02
		5.10E+02	9.28E+01			5.10E+02	9.28E+01	1.21E+03	2.81E+00
LF_gen1	PVbus	1.38E+01	-5.05E+01	4.00E+10	1.05E+10	1.73E+01	-2.59E+01	2.45E+06	-6.53E+01
		1.38E+01	-1.71E+02			1.73E+01	-1.46E+02	2.45E+06	1.75E+02
		1.38E+01	6.95E+01			1.73E+01	9.41E+01	2.45E+06	5.47E+01
LF_gen2	PVbus	1.38E+01	-4.28E+01	1.00E+09	-4.22E+08	1.32E+01	-5.06E+00	6.42E+04	-1.99E+01
		1.38E+01	-1.63E+02			1.32E+01	-1.25E+02	6.42E+04	-1.40E+02
		1.38E+01	7.72E+01			1.32E+01	1.15E+02	6.42E+04	1.00E+02
LF_gen4	PVbus	1.38E+01	3.75E+01	5.20E+10	1.59E+10	1.82E+01	6.38E+01	3.22E+06	2.05E+01
		1.38E+01	-8.25E+01			1.82E+01	-5.62E+01	3.22E+06	-9.95E+01
		1.38E+01	1.57E+02			1.82E+01	-1.76E+02	3.22E+06	1.40E+02

Table 2.8 Résumé du load-flow EMTF de la 1^{ère} simulation

La machine 3 est en sous-production par rapport à sa puissance nominale et la machine 2 consomme de la puissance réactive alors que le compensateur statique fournit du réactif afin de réguler la tension à 1.02 p.u

2.4.1.2 Référence en zone 2

Une autre simulation de load-flow est effectuée en plaçant cette fois-ci le bus Slack aux bornes de la machine 1 :

Type	Générateur 1	Générateur 2	Générateur 3	Générateur 4
Type	V δ	PV	PV	PV
Paramètres	13.8 [kV]	13.8 [kV]	13.8 [kV]	13.8 [kV]
	0 [°]	1 300 [MW]	4 400 [MW]	49 000 [MW]

Table 2.9 Paramètres de load-flow 2ème simulation

Le tableau ci-dessous est le compte-rendu de simulation effectué par EMTP-RV :

Device	Type	Vabc (kVRMSLL,deg)		P (W)	Q (VAR)	Eabc (kVRMSLL,deg)		Iabc (A,deg)	
LF_gen1	Slack	1.38E+01	2.01E-13	4.08E+10	1.08E+10	1.74E+01	2.50E+01	2.50E+06	-1.48E+01
		1.38E+01	-1.20E+02			1.74E+01	-9.50E+01	2.50E+06	-1.35E+02
		1.38E+01	1.20E+02			1.74E+01	1.45E+02	2.50E+06	1.05E+02
LF1_svc	PVbus	5.10E+02	-4.97E+00	1.15E-01	1.08E+08	5.10E+02	-4.97E+00	1.73E+02	-9.50E+01
		5.10E+02	-1.25E+02			5.10E+02	-1.25E+02	1.73E+02	1.45E+02
		5.10E+02	1.15E+02			5.10E+02	1.15E+02	1.73E+02	2.50E+01
LF_gen2	PVbus	1.38E+01	2.11E+00	1.30E+09	-2.42E+08	1.58E+01	4.35E+01	7.82E+04	1.27E+01
		1.38E+01	-1.18E+02			1.58E+01	-7.65E+01	7.82E+04	-1.07E+02
		1.38E+01	1.22E+02			1.58E+01	1.64E+02	7.82E+04	1.33E+02
LF_gen3	PVbus	1.38E+01	-4.19E+00	4.40E+09	-1.51E+08	1.70E+01	3.37E+01	2.60E+05	-2.22E+00
		1.38E+01	-1.24E+02			1.70E+01	-8.63E+01	2.60E+05	-1.22E+02
		1.38E+01	1.16E+02			1.70E+01	1.54E+02	2.60E+05	1.18E+02
LF_gen4	PVbus	1.38E+01	-2.89E+01	4.90E+10	1.52E+10	1.79E+01	-3.81E+00	3.04E+06	-4.62E+01
		1.38E+01	-1.49E+02			1.79E+01	-1.24E+02	3.04E+06	-1.66E+02
		1.38E+01	9.11E+01			1.79E+01	1.16E+02	3.04E+06	7.38E+01

Table 2.10 Résumé du load-flow EMTP de la 2^{ème} simulation

Nous voyons que toutes les machines sont à la limite de la surcharge. Un autre point à noter est que les machines de la zone 1 consomment de la puissance réactive alors que le compensateur statique fournit son maximum possible de puissance réactive.

2.4.1.3 Référence en zone 3

Une dernière simulation de load-flow est effectuée en plaçant le bus Slack aux bornes de la machine 4 :

Type	Générateur 1	Générateur 2	Générateur 3	Générateur 4
Type	PV	PV	PV	V δ
Paramètres	13.8 [kV]	13.8 [kV]	13.8 [kV]	13.8 [kV]
	42 000 [MW]	1 000 [MW]	4 000 [MW]	0 [°]

Table 2.11 Paramètres de load-flow 3^{ème} simulation

Le tableau suivant est le compte-rendu de la simulation effectué par EMTP-RV :

Device	Type	Vabc (kVRMSLL,deg)		P (W)	Q (VAR)	Eabc (kVRMSLL,deg)		Iabc (A,deg)	
LF_gen4	Slack	1.38E+01	7.78E-13	4.86E+10	1.54E+10	1.79E+01	2.49E+01	3.02E+06	-1.76E+01
		1.38E+01	-1.20E+02			1.79E+01	-9.51E+01	3.02E+06	-1.38E+02
		1.38E+01	1.20E+02			1.79E+01	1.45E+02	3.02E+06	1.02E+02
LF1_svc	PVbus	5.10E+02	4.86E+01	-5.83E-01	7.88E+08	5.10E+02	4.86E+01	1.26E+03	-4.14E+01
		5.10E+02	-7.14E+01			5.10E+02	-7.14E+01	1.26E+03	-1.61E+02
		5.10E+02	1.69E+02			5.10E+02	1.69E+02	1.26E+03	7.86E+01
LF_gen1	PVbus	1.38E+01	9.29E+01	4.20E+10	1.18E+10	1.77E+01	1.18E+02	2.58E+06	7.72E+01
		1.38E+01	-2.71E+01			1.77E+01	-1.70E+00	2.58E+06	-4.28E+01
		1.38E+01	-1.47E+02			1.77E+01	-1.22E+02	2.58E+06	-1.63E+02
LF_gen2	PVbus	1.38E+01	7.10E+01	1.00E+09	4.13E+07	1.63E+01	1.01E+02	5.92E+04	6.86E+01
		1.38E+01	-4.90E+01			1.63E+01	-1.94E+01	5.92E+04	-5.14E+01
		1.38E+01	-1.69E+02			1.63E+01	-1.39E+02	5.92E+04	-1.71E+02
LF_gen3	PVbus	1.38E+01	3.33E+01	4.00E+09	-2.36E+08	1.63E+01	6.89E+01	2.37E+05	3.67E+01
		1.38E+01	-8.67E+01			1.63E+01	-5.11E+01	2.37E+05	-8.33E+01
		1.38E+01	1.53E+02			1.63E+01	-1.71E+02	2.37E+05	1.57E+02

Table 2.12 Résumé du load-flow EMTP de la 3^{ème}

Les résultats montrent que le compensateur statique est à 75% de sa capacité maximale et que toutes les machines fournissent du réactif sauf la machine 3.

2.4.2 Temporel

60 simulations différentes sont réalisées en modifiant :

- La durée du court-circuit (12 cycles – 6 cycles) ;
- Le type de court-circuit (monophasé, biphasé, triphasé) ;
- Le lieu du court-circuit (Ligne A, B, C, D, F et aux barres B1, B2, B3, B4, B5).

Le court-circuit s’auto-éteint et les lignes ou barres correspondantes ne sont pas ouvertes.

En répétant cette opération pour chaque load-flow, un total de 180 simulations sont obtenues et le pas de simulation est de 260.4167 μ s, soit 3840 [Hz]. Cette fréquence nous est imposée par la constante de temps de la ligne la plus courte dans le système et chaque cas de simulation est simulé pendant une durée de 20 secondes.

180 simulations sont choisies en prenant soin d’avoir le plus grand nombre de cas différents ayant ainsi une grande représentativité des différents court-circuits pouvant intervenir sans toutefois être trop exhaustif car nous sommes limités en termes de ressources informatiques. En effet, en règle générale, une simulation arrivée à terme occupe un espace disque de 192 [MB], soit un total d’environ 35 [GB]. Les 180 simulations sont répertoriées à l’annexe 1. La nomenclature des fichiers est présentée comme suit :

- RpsSlackGx signifie que le bus Slack est placé à la machine x ;
- Le numéro de simulation identifiant débute à 1001. Les numéros allant de 1001 à 1060, de 1061 à 1120 et de 1121 à 1180 sont réservés pour les simulations ayant comme référence de load-flow les générateurs 1, 3 et 4 ;
- Le type de court-circuit ;
- L’endroit où surviennent le court-circuit et la longueur du court-circuit.

2.4.3 Fichier DWJ

EMTP-RV nous offre la possibilité de créer un script permettant d'automatiser les simulations. En effet, il est possible d'intervenir sur les différents attributs des éléments des simulations. Il est possible entre autre de:

- Lancer une simulation à partir du fichier ;
- Modifier le temps d'ouverture et de fermeture de chaque interrupteur ;
- Modifier la longueur de la ligne ;
- Modifier la valeur des éléments en π des lignes ou des différentes résistances, réactances et capacités.

Dans notre cas, un seul fichier DWJ est créé mais il est appliqué sur les trois fichiers de simulations en ayant au préalable changé la barre de référence et calculé le load-flow correspondant. L'annexe 3 présente le fichier DWJ utilisé :

3. Exploration des données

Dans ce chapitre, nous présenterons tout d'abord comment les simulations sont importées depuis EMTP-RV. Dans la section suivante, nous nous concentrerons sur les simulations qui ne sont pas terminées car c'est grâce à elles que nous pourrions définir les critères de discrimination entre les simulations stables et instables.

Par la suite, un sous-chapitre expliquera comment les nouvelles variables sont créées à partir de celles recueillies par EMTP-RV et comment les sous-ensembles d'apprentissage de test et de validation sont créés.

Finalement, nous présenterons comment les données sont exportées au format .CSV vers le logiciel KNIME.

3.1 Importations des données sous MATLAB

EMTP-RV nous permet d'exporter les données dans un format ASCII (.txt) ou dans un format MATLAB (.mat). Ce dernier est choisi car MATLAB permet de traiter les grandes matrices de données en simple précision et par conséquent, diviser par deux la taille de l'espace en mémoire et dans le disque dur.

En effet, pour chaque simulation, le fichier exporté est une matrice de 76 800 individus de long, soit 3 840 points par secondes durant 20 secondes et comme expliqué au paragraphe 2.3.1.6, 164 colonnes, soit 156 mesures électriques et une vitesse de rotation et un angle de charge pour chaque machine.

À l'importation du fichier .mat de base, le workspace de MATLAB contient une cellule :

- Y contient 164 variables et 76 800 individus ;
- $Ylabels$ contient le nom de la variable associée à la cellule Y ;
- X contient 164 variables de temps et 76 800 individus ;
- $Xlabels$ contient 164 noms de variable temps

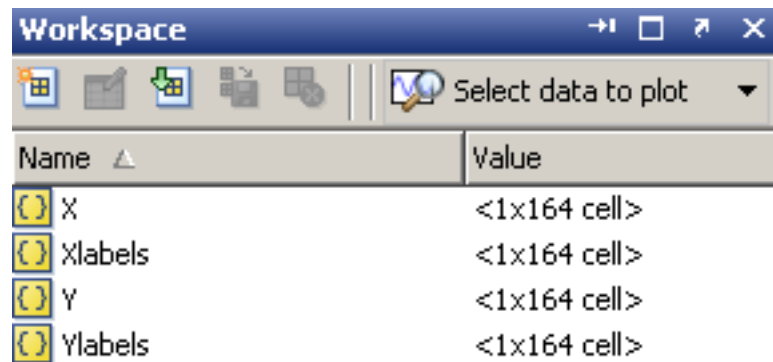
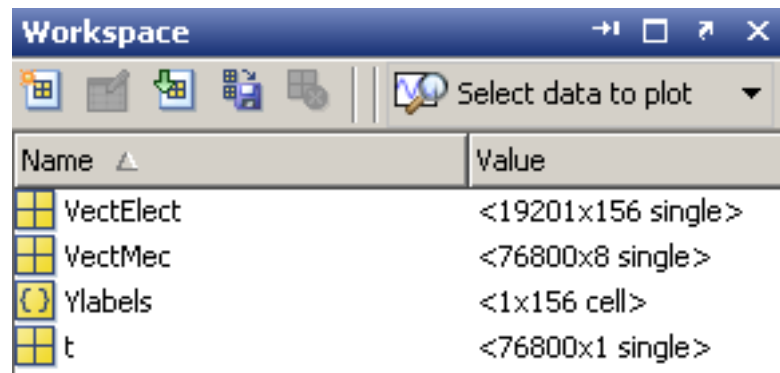


Figure 3.1 Workspace MATLAB après importation

À partir de la cellule X , nous créons un vecteur temps t et les autres 163 colonnes de la cellule X et la cellule $Xlabels$ sont supprimées.

La cellule Y est convertie du format double au format simple, puis elle est séparée en une matrice mécanique qui contient les angles et les vitesses des quatre machines et une matrice électrique qui contient les variables mesurées.

Finalement, par soucis de disque dur et de mémoire vive, nous ne garderons que cinq secondes du vecteur électrique réduisant ainsi son nombre d'individus à un maximum de 19201.



The screenshot shows the MATLAB Workspace window with a toolbar at the top containing icons for workspace, command window, script editor, and a 'Select data to plot' dropdown. Below the toolbar is a table with two columns: 'Name' and 'Value'.

Name	Value
VectElect	<19201x156 single>
VectMec	<76800x8 single>
Ylabels	<1x156 cell>
t	<76800x1 single>

Figure 3.2 Séparation de X en $VectElect$ et $VectMec$

Ces variables écrasent l'ancien fichier .mat réduisant par conséquent sa taille à 9 [MB] par fichier, soit 1.6 [GB] pour l'ensemble des simulations.

Par après, la matrice $VectElect$ est séparée en quatre sous-matrices correspondant au générateur associé soit : $VectElectGen1$, $VectElectGen2$, $VectElectGen3$ et $VectElectGen4$. et $Ylabels$ en $LabelsGen1$, $LabelsGen2$, $LabelsGen3$ et $LabelsGen4$.

3.2 Simulations non-terminées

Les simulations présentées à la Table 3.1, EMTP-RV a jugé bon d'arrêter la simulation avant la fin car les vitesses des machines ne convergeaient pas dans un état stable et par conséquent, le nombre d'individu de ces machines est réduit et donc leur taille dans le disque dur aussi. Étant donné que le court-circuit intervient à une seconde et dure 0.2 secondes pour les courts-circuit longs et 0.1 secondes pour les courts-circuits courts, les simulations surlignées en bleu seront retirées car elles s'arrêtent avant la fin du court-circuit et étant donné que l'arbre de décision sera entraîné avec les données après que le court-circuit soit arrêté, ces simulations n'apportent rien en termes de connaissances.

Nom de la simulation	Nombre d'individus	Fin de simulation
'rpsslackg1_1011_tripahse_barreb1_long.mat'	4232	1.10
'rpsslackg1_1012_tripahse_barreb1_court.mat'	4232	1.10
'rpsslackg1_1017_tripahse_barreb4_long.mat'	23173	6.03
'rpsslackg1_1019_tripahse_barreb5_long.mat'	22248	5.79
'rpsslackg1_1048_monophase_ligned_30_court.mat'	21617	5.63
'rpsslackg1_1051_monophase_barreb1_long.mat'	4232	1.10
'rpsslackg1_1052_monophase_barreb1_court.mat'	4232	1.10
'rpsslackg1_1053_monophase_barreb2_long.mat'	21213	5.52
'rpsslackg1_1054_monophase_barreb2_court.mat'	49968	13.01
'rpsslackg1_1057_monophase_barreb4_court.mat'	4400	1.15
'rpsslackg1_1058_monophase_barreb4_long.mat'	55902	14.56
'rpsslackg1_1059_monophase_barreb5_long.mat'	18451	4.80
'rpsslackg1_1060_monophase_barreb5_court.mat'	16165	4.21
'rpsslackg3_1072_tripahse_barreb1_court.mat'	53710	13.99
'rpsslackg3_1080_tripahse_barreb5_court.mat'	60360	15.72
'rpsslackg3_1099_biphase_barreb5_long.mat'	29859	7.78
'rpsslackg3_1109_monophase_ligne_f_long.mat'	58908	15.34
'rpsslackg3_1111_monophase_barreb1_long.mat'	5921	1.54
'rpsslackg3_1119_monophase_barreb5_long.mat'	17740	4.62
'rpsslackg3_1120_monophase_barreb5_court.mat'	59167	15.41
'rpsslackg4_1131_tripahse_barreb1_long.mat'	29125	7.58
'rpsslackg4_1137_tripahse_barreb4_long.mat'	48027	12.51
'rpsslackg4_1151_biphase_barreb1_long.mat'	42774	11.14
'rpsslackg4_1159_biphase_barreb5_long.mat'	56217	14.64
'rpsslackg4_1162_monophase_ligne_a_court.mat'	54491	14.19
'rpsslackg4_1172_monophase_barreb1_court.mat'	41664	10.85
'rpsslackg4_1178_monophase_barreb4_court.mat'	58004	15.11
'rpsslackg4_1180_monophase_barreb5_court.mat'	15074	3.93

Table 3.1 Simulations finies avant 20 secondes

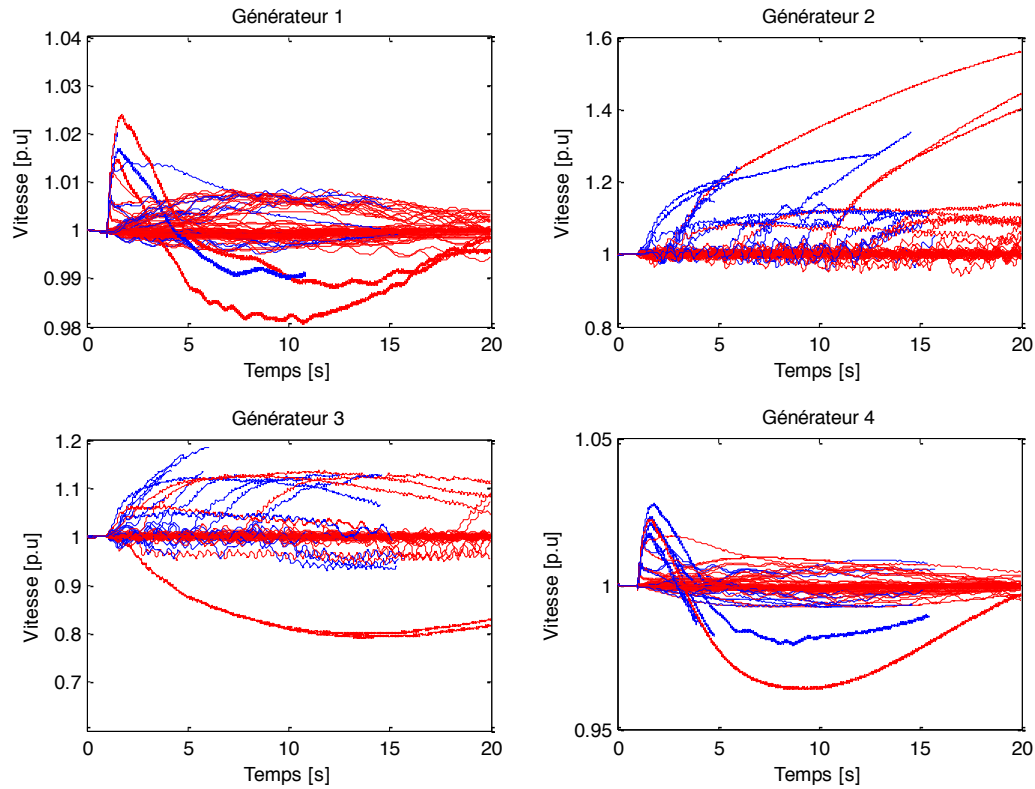


Figure 3.3 Caractéristique des vitesses. En bleu les simulations arrêtées avant 20 sec

La figure précédente reporte les 180 simulations et surligne en bleu celle de la Table 3.1. Nous voyons aussi que bien des cas sont instables sans pour autant que EMTP-RV n'arrête la simulation.

3.3 Tri des simulations stables et instables

3.3.1 Introduction théorique au centre d'inertie

Après une perturbation, une, ou un groupe de machines, tend soit à accélérer, soit à décélérer ou à revenir à un point stable après une période transitoire dépendamment de la configuration du réseau et du niveau de charge post-contingence. Toute la problématique réside dans la discrimination des machines qui ont perdu leur stabilité et à les isoler du reste du réseau.

Avec la formule suivante, selon [7] et [20], il est possible de définir une base commune ou une moyenne de la vitesse et de l'angle de toutes les machines du système appelée centre d'inertie et définie par la formule suivante :

$$\overline{\delta_{coi}} = \frac{1}{M_T} \cdot \sum_{j=1}^r M_j \cdot \delta_j \quad (3.1)$$

Cette formule ci-dessus n'est qu'une moyenne de tous les angles des j machines pondérées avec leur moment d'inertie M_j . M_T est la somme de tous les moments d'inertie de toutes les machines et r est le nombre total de zone dans le système.

δ_j est la moyenne de l'angle δ_i des N machines dans la même zone de production j , et est défini par :

$$\overline{\delta_j} = \frac{1}{N} \cdot \sum_{i=1}^N \delta_i \quad (3.2)$$

Finalement, le centre d'inertie COI de l'angle et la différence des deux formules précédente, soit :

$$\delta_j^{COI} = \overline{\delta_j} - \overline{\delta_{coi}} \quad (3.3)$$

De la même façon, il est possible de déterminer le centre d'inertie de la vitesse des différentes machines avec les trois formules suivantes :

$$\overline{\omega_{col}} = \frac{1}{M_T} \cdot \sum_{j=1}^r M_j \cdot \omega_j \quad (3.4)$$

$$\overline{\omega_j} = \frac{1}{N} \cdot \sum_{i=1}^N \omega_i \quad (3.5)$$

$$\omega_j^{COI} = \overline{\omega_j} - \overline{\omega_{col}} \quad (3.6)$$

3.3.2 Critères d'instabilité

Selon [7], il est possible de déterminer si une machine est stable ou instable en vérifiant les critères suivant :

- Si, pour une machine, le centre d'inertie d'angle est compris entre +/- pi ET que son centre d'inertie de vitesse est lent, alors la machine est stable ;
- Si, pour une machine, le centre d'inertie d'angle est supérieur à +/- pi ET que son centre d'inertie de vitesse est rapide, alors la machine est instable ;

Le problème dans l'application de ces critères est de déterminer le seuil minimal de vitesse à partir duquel une machine est considérée comme instable, ce qui revient à quantifier numériquement rapide et faible.

Sachant que si EMTP-RV a arrêté une simulation avant la fin, celle-ci est considérée comme divergente. En utilisant ce critère, il est possible de prendre le minimum du COI de vitesse de ces simulations et de le déterminer comme seuil minimum de discrimination des cas stables et instables.

Les figures suivantes montrent la valeur du centre d'inertie d'angle et de vitesse des machines pour les simulations qui ont été arrêtées avant 20 secondes.

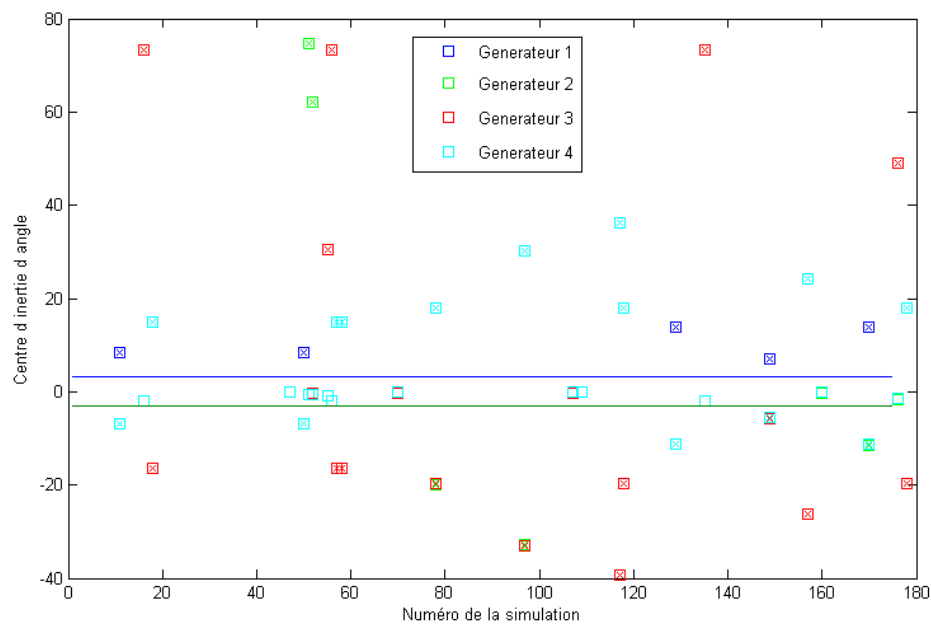


Figure 3.4 COIA des simulations pour $t < 20[s]$

Les deux barres bleues et vertes dans le graphique ci-dessus indiquent ± 3.14 [rad].

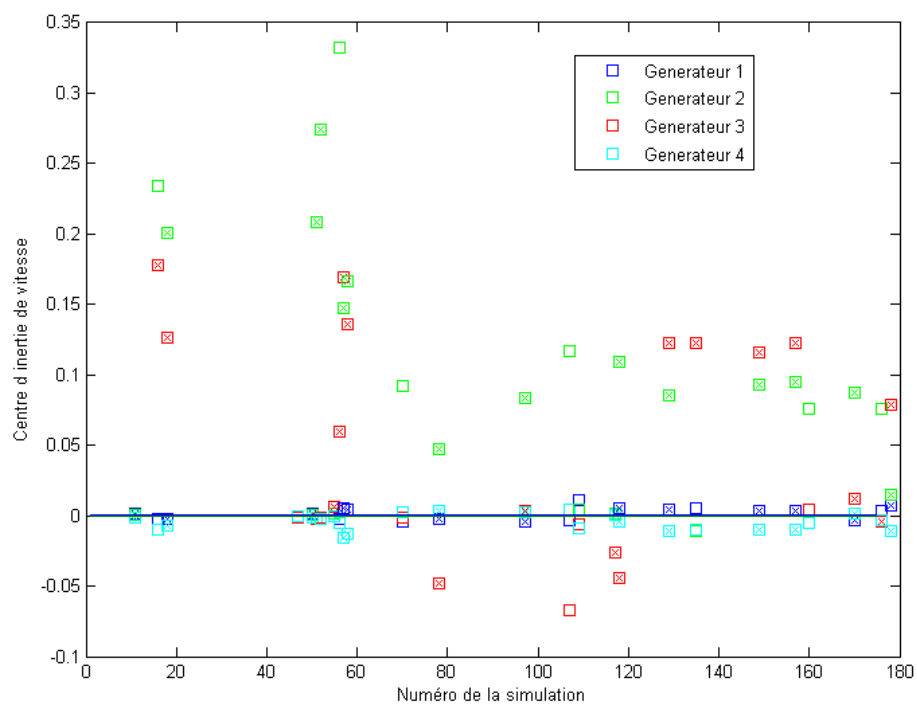


Figure 3.5 COI Ω des simulations pour $t < 20[s]$



La figure suivante est celle qui nous permet de définir le seuil. En effet, après avoir marqué d'un X les simulations dont le $COI\Omega$ était supérieur à π , nous cherchons la simulation sur le graphique du $COI\Omega$ la plus faible valeur de centre d'inertie.

Il s'avère qu'il s'agit des simulations 11 et 50 avec le générateur 2 qui fixent le seuil minimum de vitesse à 0.0002107 [p.u]. Un seuil un peu plus conservatif de ± 0.0002 [p.u] est choisi pour la suite et est affiché avec la barre bleue et verte :

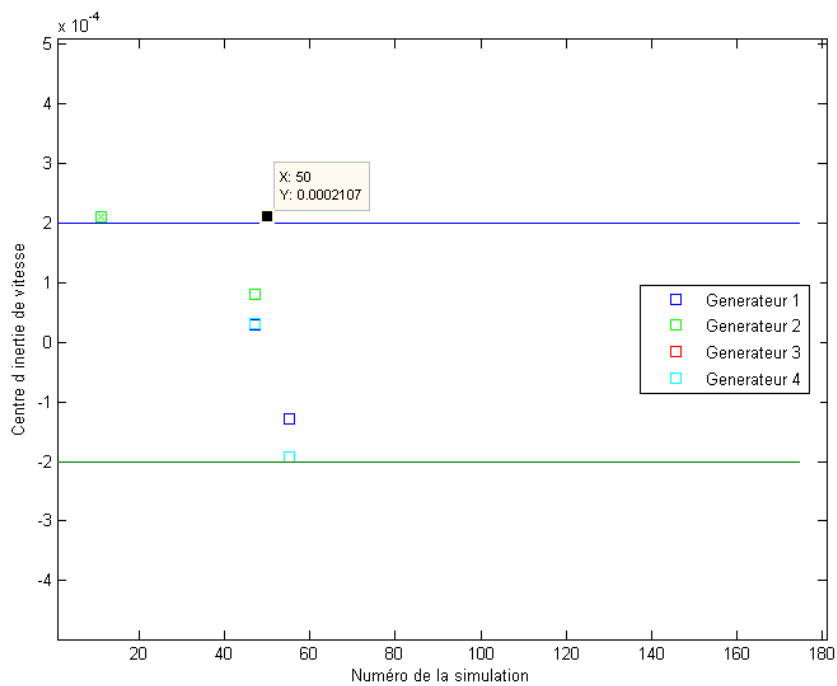


Figure 3.6 Agrandissement de $COI\Omega$ pour trouver le seuil

3.3.3 Application sur l'ensemble de données

Ayant en main le seuil minimum de centre d'inertie de vitesse : ± 0.0002 [p.u] et d'angle ± 3.14 [rad], nous appliquons ces deux conditions sur l'ensemble des simulations. Les figures suivantes montrent les simulations stables (carré vide) et instables (carré avec une croix)

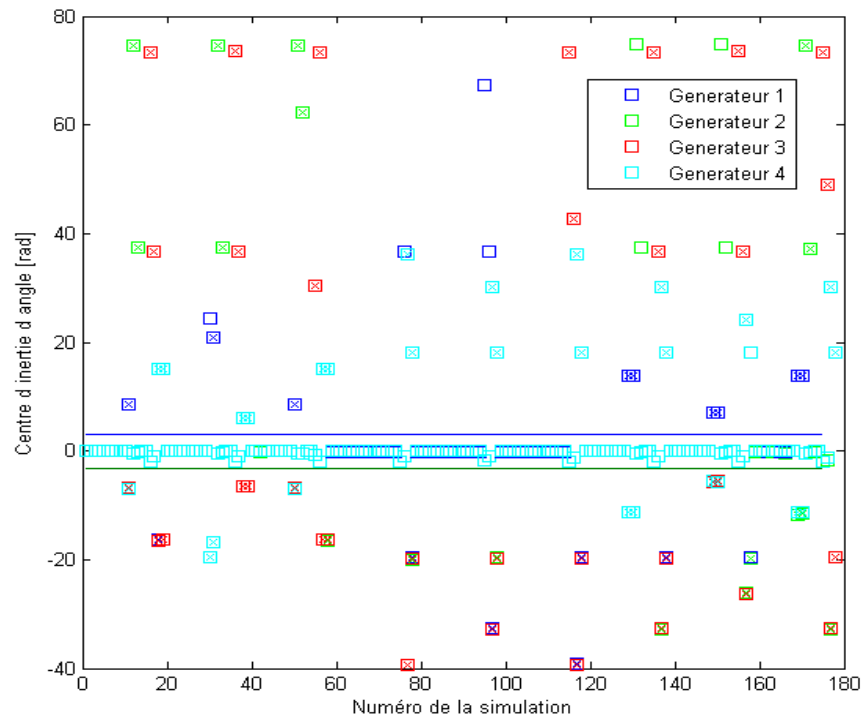


Figure 3.7 COIA de toutes les simulations

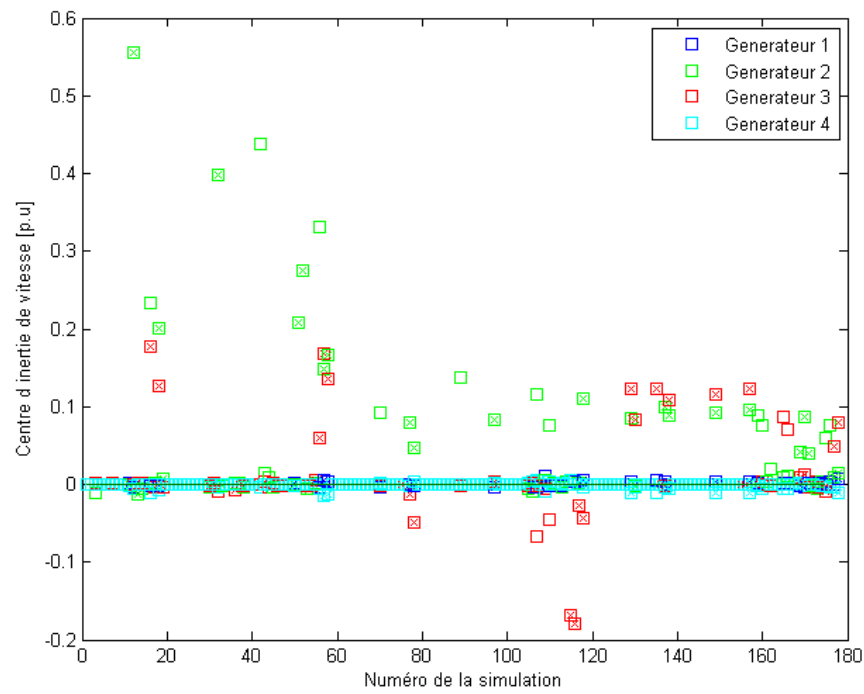


Figure 3.8 COIΩ de toutes les simulations

Le tableau suivant présente le nombre de cas stables et instables pour chaque générateur. Nous voyons que les machines 1 et 4 sont plus stables que les machines 2 et 3.

	Générateur 1	Générateur 2	Générateur 3	Générateur 4
Stables	152	143	136	151
Instables	26	35	42	27

Table 3.2 Répartition des simulations stables et instables

Gen 1	Gen 2	Gen 3	Gen 4	Gen 1	Gen 2	Gen 3	Gen 4
1012	1012	1012	1012	1100	1100	1100	1100
	1013					1117	
	1014					1118	
		1017		1119	1119	1119	1119
		1018		1120	1120	1120	1120
1019	1019	1019	1019	1131	1131	1131	1131
1020	1020	1020	1020	1132	1132	1132	1132
	1031	1031	1031			1137	
1032	1032	1032	1032			1138	
	1033			1139	1139	1139	1139
	1034			1140	1140	1140	1140
		1037		1151	1151	1151	1151
		1038		1152		1152	1152
1039	1039	1039	1039			1157	
1040	1040	1040	1040			1158	
1052	1052	1052	1052	1159	1159	1159	1159
	1053				1160		
	1054			1171	1171	1171	1171
		1057		1172	1172	1172	1172
		1058			1173		
1059	1059	1059	1059		1174		
1060	1060	1060	1060			1177	
		1077				1178	
1079	1079	1079	1079	1179	1179	1179	1179
1080	1080	1080	1080	1180	1180	1180	1180
1099	1099	1099	1099				

Table 3.3 Numéro de simulations des machines ayant divergé

En analysant le tableau suivant, parmi les 51 simulations ayant divergé nous voyons que :

- Dans 25 simulations, tous le système est instable ;
- Dans 2 simulations, 3 machines ont divergé en même temps ;
- Dans 24 simulations, 1 machine a divergé.

Notons que lorsque le bus de référence est placé sur le générateur 1 (simulations de 1001 à 1060) :

- 24 simulations divergent pour un total de 59 machines divergentes ;
- Dans 12 simulations, une seule machine diverge ;
- Dans 9 simulations, tout le réseau est instable.

Si le bus de référence est placé dans le générateur 3 (simulations de 1061 à 1120) :

- 9 simulations divergent pour un total de 27 machines divergentes ;
- Dans 3 simulations, une machine diverge seule ;
- Dans 6 simulations, c'est tout le système qui est instable.

Finalement. lorsque le bus de référence est placé dans le générateur 4 (simulations de 1121 à 1180) :

- 20 simulations divergent pour un total de 52 machines ;
- Dans 9 simulations, une machine diverge seule ;
- Dans 10 simulations, c'est tout le réseau qui est instable,

Quand on lit ces résultats, nous pouvons voir qu'une seule machine diverge, soit c'est tout le système qui diverge. De plus, lorsque la barre de référence est placée dans les générateurs 1 et 4, il y a un plus grand nombre de cas divergents.

3.3.4 Performances de discrimination

Après avoir déterminé quelle simulation diverge, la figure suivante nous permet de visualiser la qualité de la discrimination :

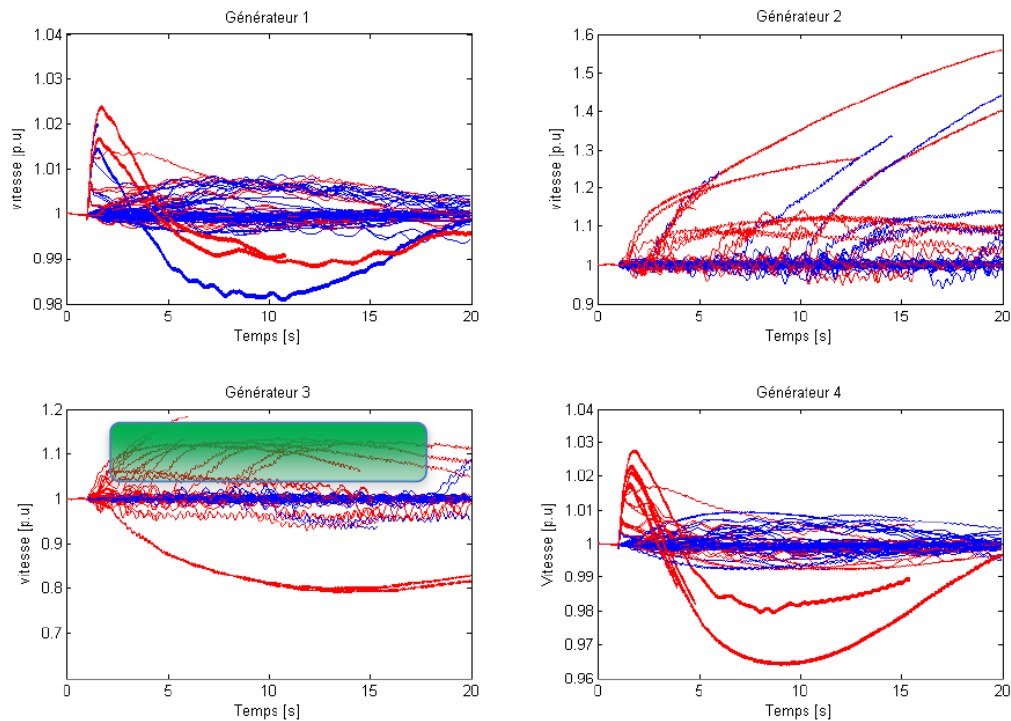


Figure 3.9 Vitesse des 180 générateurs : Stable Rouge, Instable en Bleu

Dans la figure verte, nous voyons que le générateur accélère jusqu'à une fréquence de 66 [Hz] sans pour autant que le générateur soit déclaré instable. La même déduction s'applique au générateur 2. Cette méthode, bien que facile à mettre en œuvre, peut être améliorée. RTE, l'entreprise chargée du réseau très haute tension en France, utilise comme critères [21] vingt inversions de puissance ou quatre tours d'angle interne. Si l'une de ces deux conditions est vérifiée, la machine est déclarée instable. Une étude cherchant à avoir la meilleure discrimination peut être d'une grande utilité. Les tableaux ci-dessous relèvent les simulations qui ont été classées comme instables par cette méthode. Les cases coloriées correspondent à une simulation divergente.

Générateur 1	
'rpsslackg1_1001_tripahse_lignea_30_long.mat'	'rpsslackg1_1031_biphase_barreb1_long.mat'
'rpsslackg1_1002_tripahse_lignea_30_court.mat'	'rpsslackg1_1032_biphase_barreb1_court.mat'
'rpsslackg1_1003_tripahse_ligne_b_30_long.mat'	'rpsslackg1_1033_biphase_barreb2_long.mat'
'rpsslackg1_1004_tripahse_ligne_b_30_court.mat'	'rpsslackg1_1034_biphase_barreb2_court.mat'
'rpsslackg1_1005_tripahse_lignec_30_long.mat'	'rpsslackg1_1035_biphase_barreb3_long.mat'
'rpsslackg1_1006_tripahse_lignec_30_court.mat'	'rpsslackg1_1036_biphase_barreb3_court.mat'
'rpsslackg1_1007_tripahse_ligned_30_long.mat'	'rpsslackg1_1037_biphase_barreb4_long.mat'
'rpsslackg1_1008_tripahse_ligned_30_court.mat'	'rpsslackg1_1038_biphase_barreb4_court.mat'
'rpsslackg1_1009_tripahse_lignef_30_court.mat'	'rpsslackg1_1039_biphase_barreb5_long.mat'
'rpsslackg1_1010_tripahse_lignef_court.mat'	'rpsslackg1_1040_biphase_barreb5_court.mat'
	'rpsslackg1_1041_monophase_lignea_30_long.mat'
'rpsslackg1_1012_tripahse_barreb1_court.mat'	'rpsslackg1_1042_monophase_lignea_30_court.mat'
'rpsslackg1_1013_tripahse_barreb2_long.mat'	'rpsslackg1_1043_monophase_ligne_b_30_long.mat'
'rpsslackg1_1014_tripahse_barreb2_court.mat'	'rpsslackg1_1044_monophase_ligne_b_30_court.mat'
'rpsslackg1_1015_tripahse_barreb3_long.mat'	'rpsslackg1_1045_monophase_lignec_30_long.mat'
'rpsslackg1_1016_tripahse_barreb3_court.mat'	'rpsslackg1_1046_monophase_lignec_30_court.mat'
'rpsslackg1_1017_tripahse_barreb4_long.mat'	'rpsslackg1_1047_monophase_ligned_30_long.mat'
'rpsslackg1_1018_tripahse_barreb4_court.mat'	'rpsslackg1_1048_monophase_ligned_30_court.mat'
'rpsslackg1_1019_tripahse_barreb5_long.mat'	'rpsslackg1_1049_monophase_lignef_30_long.mat'
'rpsslackg1_1020_tripahse_barreb5_court.mat'	'rpsslackg1_1050_monophase_lignef_30_court.mat'
'rpsslackg1_1021_biphase_lignea_30_long.mat'	
'rpsslackg1_1022_biphase_lignea_30_court.mat'	'rpsslackg1_1052_monophase_barreb1_court.mat'
'rpsslackg1_1023_biphase_ligne_b_30_long.mat'	'rpsslackg1_1053_monophase_barreb2_long.mat'
'rpsslackg1_1024_biphase_ligne_b_30_court.mat'	'rpsslackg1_1054_monophase_barreb2_court.mat'
'rpsslackg1_1025_biphase_lignec_30_long.mat'	'rpsslackg1_1055_monophase_barreb3_long.mat'
'rpsslackg1_1026_biphase_lignec_30_court.mat'	'rpsslackg1_1056_monophase_barreb3_court.mat'
'rpsslackg1_1027_biphase_ligned_30_long.mat'	'rpsslackg1_1057_monophase_barreb4_court.mat'
'rpsslackg1_1028_biphase_ligned_30_court.mat'	'rpsslackg1_1058_monophase_barreb4_long.mat'
'rpsslackg1_1029_biphase_lignef_30_long.mat'	'rpsslackg1_1059_monophase_barreb5_long.mat'
'rpsslackg1_1030_biphase_lignef_30_court.mat'	'rpsslackg1_1060_monophase_barreb5_court.mat'

Table 3.4 Tableau des simulations divergentes selon le COI pour le générateur 1

Générateur 1	
'rpsslackg3_1061_tripahse_lignea_long.mat'	'rpsslackg3_1091_biphase_barreb1_long.mat'
'rpsslackg3_1062_tripahse_lignea_court.mat'	'rpsslackg3_1092_biphase_barreb1_court.mat'
'rpsslackg3_1063_tripahse_ligne_b_long.mat'	'rpsslackg3_1093_biphase_barreb2_long.mat'
'rpsslackg3_1064_tripahse_ligne_b_court.mat'	'rpsslackg3_1094_biphase_barreb2_court.mat'
'rpsslackg3_1065_tripahse_lignec_long.mat'	'rpsslackg3_1095_biphase_barreb3_long.mat'
'rpsslackg3_1066_tripahse_lignec_court.mat'	'rpsslackg3_1096_biphase_barreb3_court.mat'
'rpsslackg3_1067_tripahse_ligned_long.mat'	'rpsslackg3_1097_biphase_barreb4_long.mat'
'rpsslackg3_1068_tripahse_ligned_court.mat'	'rpsslackg3_1098_biphase_barreb4_court.mat'
'rpsslackg3_1069_tripahse_lignef_long.mat'	'rpsslackg3_1099_biphase_barreb5_long.mat'
'rpsslackg3_1070_tripahse_lignef_court.mat'	'rpsslackg3_1100_biphase_barreb5_court.mat'
'rpsslackg3_1071_tripahse_barreb1_long.mat'	'rpsslackg3_1101_monophase_lignea_long.mat'
'rpsslackg3_1072_tripahse_barreb1_court.mat'	'rpsslackg3_1102_monophase_lignea_court.mat'
'rpsslackg3_1073_tripahse_barreb2_long.mat'	'rpsslackg3_1103_monophase_ligne_b_long.mat'
'rpsslackg3_1074_tripahse_barreb2_court.mat'	'rpsslackg3_1104_monophase_ligne_b_court.mat'
'rpsslackg3_1075_tripahse_barreb3_long.mat'	'rpsslackg3_1105_monophase_lignec_long.mat'
'rpsslackg3_1076_tripahse_barreb3_court.mat'	'rpsslackg3_1106_monophase_lignec_court.mat'
'rpsslackg3_1077_tripahse_barreb4_long.mat'	'rpsslackg3_1107_monophase_ligned_long.mat'
'rpsslackg3_1078_tripahse_barreb4_court.mat'	'rpsslackg3_1108_monophase_ligned_court.mat'
'rpsslackg3_1079_tripahse_barreb5_long.mat'	'rpsslackg3_1109_monophase_lignef_long.mat'
'rpsslackg3_1080_tripahse_barreb5_court.mat'	'rpsslackg3_1110_monophase_lignef_court.mat'
'rpsslackg3_1081_biphase_lignea_long.mat'	'rpsslackg3_1111_monophase_barreb1_long.mat'
'rpsslackg3_1082_biphase_lignea_court.mat'	'rpsslackg3_1112_monophase_barreb1_court.mat'
'rpsslackg3_1083_biphase_ligne_b_long.mat'	'rpsslackg3_1113_monophase_barreb2_long.mat'
'rpsslackg3_1084_biphase_ligne_b_court.mat'	'rpsslackg3_1114_monophase_barreb2_court.mat'
'rpsslackg3_1085_biphase_lignec_long.mat'	'rpsslackg3_1115_monophase_barreb3_long.mat'
'rpsslackg3_1086_biphase_lignec_court.mat'	'rpsslackg3_1116_monophase_barreb3_court.mat'
'rpsslackg3_1087_biphase_ligned_long.mat'	'rpsslackg3_1117_monophase_barreb4_long.mat'
'rpsslackg3_1088_biphase_ligned_court.mat'	'rpsslackg3_1118_monophase_barreb4_court.mat'
'rpsslackg3_1089_biphase_lignef_long.mat'	'rpsslackg3_1119_monophase_barreb5_long.mat'
'rpsslackg3_1090_biphase_lignef_court.mat'	'rpsslackg3_1120_monophase_barreb5_court.mat'

Table 3.5 Tableau des simulations divergentes selon le COI pour le générateur 1

Générateur 1	
'rpsslackg4_1121_tripahse_lignea_long.mat'	'rpsslackg4_1151_biphase_barreb1_long.mat'
'rpsslackg4_1122_tripahse_lignea_court.mat'	'rpsslackg4_1152_biphase_barreb1_court.mat'
'rpsslackg4_1123_tripahse_ligne_b_long.mat'	'rpsslackg4_1153_biphase_barreb2_long.mat'
'rpsslackg4_1124_tripahse_ligne_b_court.mat'	'rpsslackg4_1154_biphase_barreb2_court.mat'
'rpsslackg4_1125_tripahse_lignec_long.mat'	'rpsslackg4_1155_biphase_barreb3_long.mat'
'rpsslackg4_1126_tripahse_lignec_court.mat'	'rpsslackg4_1156_biphase_barreb3_court.mat'
'rpsslackg4_1127_tripahse_ligned_long.mat'	'rpsslackg4_1157_biphase_barreb4_long.mat'
'rpsslackg4_1128_tripahse_ligned_court.mat'	'rpsslackg4_1158_biphase_barreb4_court.mat'
'rpsslackg4_1129_tripahse_ligne_f_long.mat'	'rpsslackg4_1159_biphase_barreb5_long.mat'
'rpsslackg4_1130_tripahse_ligne_f_court.mat'	'rpsslackg4_1160_biphase_barreb5_court.mat'
'rpsslackg4_1131_tripahse_barreb1_long.mat'	'rpsslackg4_1161_monophase_lignea_long.mat'
'rpsslackg4_1132_tripahse_barreb1_court.mat'	'rpsslackg4_1162_monophase_lignea_court.mat'
'rpsslackg4_1133_tripahse_barreb2_long.mat'	'rpsslackg4_1163_monophase_ligne_b_long.mat'
'rpsslackg4_1134_tripahse_barreb2_court.mat'	'rpsslackg4_1164_monophase_ligne_b_court.mat'
'rpsslackg4_1135_tripahse_barreb3_long.mat'	'rpsslackg4_1165_monophase_lignec_long.mat'
'rpsslackg4_1136_tripahse_barreb3_court.mat'	'rpsslackg4_1166_monophase_lignec_court.mat'
'rpsslackg4_1137_tripahse_barreb4_long.mat'	'rpsslackg4_1167_monophase_ligned_long.mat'
'rpsslackg4_1138_tripahse_barreb4_court.mat'	'rpsslackg4_1168_monophase_ligned_court.mat'
'rpsslackg4_1139_tripahse_barreb5_long.mat'	'rpsslackg4_1169_monophase_ligne_f_long.mat'
'rpsslackg4_1140_tripahse_barreb5_court.mat'	'rpsslackg4_1170_monophase_ligne_f_court.mat'
'rpsslackg4_1141_biphase_lignea_long.mat'	'rpsslackg4_1171_monophase_barreb1_long.mat'
'rpsslackg4_1142_biphase_lignea_court.mat'	'rpsslackg4_1172_monophase_barreb1_court.mat'
'rpsslackg4_1143_biphase_ligne_b_long.mat'	'rpsslackg4_1173_monophase_barreb2_long.mat'
'rpsslackg4_1144_biphase_ligne_b_court.mat'	'rpsslackg4_1174_monophase_barreb2_court.mat'
'rpsslackg4_1145_biphase_lignec_long.mat'	'rpsslackg4_1175_monophase_barreb3_long.mat'
'rpsslackg4_1146_biphase_lignec_court.mat'	'rpsslackg4_1176_monophase_barreb3_court.mat'
'rpsslackg4_1147_biphase_ligned_long.mat'	'rpsslackg4_1177_monophase_barreb4_long.mat'
'rpsslackg4_1148_biphase_ligned_court.mat'	'rpsslackg4_1178_monophase_barreb4_court.mat'
'rpsslackg4_1149_biphase_ligne_f_long.mat'	'rpsslackg4_1179_monophase_barreb5_long.mat'
'rpsslackg4_1150_biphase_ligne_f_court.mat'	'rpsslackg4_1180_monophase_barreb5_court.mat'

Table 3.6 Tableau des simulations divergentes selon le COI pour le générateur 1

Générateur 2	
'rpsslackg1_1001_tripahse_ligne_a_30_long.mat'	'rpsslackg1_1031_biphase_barreb1_long.mat'
'rpsslackg1_1002_tripahse_ligne_a_30_court.mat'	'rpsslackg1_1032_biphase_barreb1_court.mat'
'rpsslackg1_1003_tripahse_ligne_b_30_long.mat'	'rpsslackg1_1033_biphase_barreb2_long.mat'
'rpsslackg1_1004_tripahse_ligne_b_30_court.mat'	'rpsslackg1_1034_biphase_barreb2_court.mat'
'rpsslackg1_1005_tripahse_ligne_c_30_long.mat'	'rpsslackg1_1035_biphase_barreb3_long.mat'
'rpsslackg1_1006_tripahse_ligne_c_30_court.mat'	'rpsslackg1_1036_biphase_barreb3_court.mat'
'rpsslackg1_1007_tripahse_ligne_d_30_long.mat'	'rpsslackg1_1037_biphase_barreb4_long.mat'
'rpsslackg1_1008_tripahse_ligne_d_30_court.mat'	'rpsslackg1_1038_biphase_barreb4_court.mat'
'rpsslackg1_1009_tripahse_ligne_f_30_court.mat'	'rpsslackg1_1039_biphase_barreb5_long.mat'
'rpsslackg1_1010_tripahse_ligne_f_court.mat'	'rpsslackg1_1040_biphase_barreb5_court.mat'
	'rpsslackg1_1041_monophase_ligne_a_30_long.mat'
'rpsslackg1_1012_tripahse_barreb1_court.mat'	'rpsslackg1_1042_monophase_ligne_a_30_court.mat'
'rpsslackg1_1013_tripahse_barreb2_long.mat'	'rpsslackg1_1043_monophase_ligne_b_30_long.mat'
'rpsslackg1_1014_tripahse_barreb2_court.mat'	'rpsslackg1_1044_monophase_ligne_b_30_court.mat'
'rpsslackg1_1015_tripahse_barreb3_long.mat'	'rpsslackg1_1045_monophase_ligne_c_30_long.mat'
'rpsslackg1_1016_tripahse_barreb3_court.mat'	'rpsslackg1_1046_monophase_ligne_c_30_court.mat'
'rpsslackg1_1017_tripahse_barreb4_long.mat'	'rpsslackg1_1047_monophase_ligne_d_30_long.mat'
'rpsslackg1_1018_tripahse_barreb4_court.mat'	'rpsslackg1_1048_monophase_ligne_d_30_court.mat'
'rpsslackg1_1019_tripahse_barreb5_long.mat'	'rpsslackg1_1049_monophase_ligne_f_30_long.mat'
'rpsslackg1_1020_tripahse_barreb5_court.mat'	'rpsslackg1_1050_monophase_ligne_f_30_court.mat'
'rpsslackg1_1021_biphase_ligne_a_30_long.mat'	
'rpsslackg1_1022_biphase_ligne_a_30_court.mat'	'rpsslackg1_1052_monophase_barreb1_court.mat'
'rpsslackg1_1023_biphase_ligne_b_30_long.mat'	'rpsslackg1_1053_monophase_barreb2_long.mat'
'rpsslackg1_1024_biphase_ligne_b_30_court.mat'	'rpsslackg1_1054_monophase_barreb2_court.mat'
'rpsslackg1_1025_biphase_ligne_c_30_long.mat'	'rpsslackg1_1055_monophase_barreb3_long.mat'
'rpsslackg1_1026_biphase_ligne_c_30_court.mat'	'rpsslackg1_1056_monophase_barreb3_court.mat'
'rpsslackg1_1027_biphase_ligne_d_30_long.mat'	'rpsslackg1_1057_monophase_barreb4_court.mat'
'rpsslackg1_1028_biphase_ligne_d_30_court.mat'	'rpsslackg1_1058_monophase_barreb4_long.mat'
'rpsslackg1_1029_biphase_ligne_f_30_long.mat'	'rpsslackg1_1059_monophase_barreb5_long.mat'
'rpsslackg1_1030_biphase_ligne_f_30_court.mat'	'rpsslackg1_1060_monophase_barreb5_court.mat'

Table 3.7 Tableau des simulations divergentes selon le COI pour le générateur 2

Générateur 2	
'rpsslackg3_1061_tripahse_lignea_long.mat'	'rpsslackg3_1091_biphase_barreb1_long.mat'
'rpsslackg3_1062_tripahse_lignea_court.mat'	'rpsslackg3_1092_biphase_barreb1_court.mat'
'rpsslackg3_1063_tripahse_ligne_b_long.mat'	'rpsslackg3_1093_biphase_barreb2_long.mat'
'rpsslackg3_1064_tripahse_ligne_b_court.mat'	'rpsslackg3_1094_biphase_barreb2_court.mat'
'rpsslackg3_1065_tripahse_lignec_long.mat'	'rpsslackg3_1095_biphase_barreb3_long.mat'
'rpsslackg3_1066_tripahse_lignec_court.mat'	'rpsslackg3_1096_biphase_barreb3_court.mat'
'rpsslackg3_1067_tripahse_ligned_long.mat'	'rpsslackg3_1097_biphase_barreb4_long.mat'
'rpsslackg3_1068_tripahse_ligned_court.mat'	'rpsslackg3_1098_biphase_barreb4_court.mat'
'rpsslackg3_1069_tripahse_lignef_long.mat'	'rpsslackg3_1099_biphase_barreb5_long.mat'
'rpsslackg3_1070_tripahse_lignef_court.mat'	'rpsslackg3_1100_biphase_barreb5_court.mat'
'rpsslackg3_1071_tripahse_barreb1_long.mat'	'rpsslackg3_1101_monophase_lignea_long.mat'
'rpsslackg3_1072_tripahse_barreb1_court.mat'	'rpsslackg3_1102_monophase_lignea_court.mat'
'rpsslackg3_1073_tripahse_barreb2_long.mat'	'rpsslackg3_1103_monophase_ligne_b_long.mat'
'rpsslackg3_1074_tripahse_barreb2_court.mat'	'rpsslackg3_1104_monophase_ligne_b_court.mat'
'rpsslackg3_1075_tripahse_barreb3_long.mat'	'rpsslackg3_1105_monophase_lignec_long.mat'
'rpsslackg3_1076_tripahse_barreb3_court.mat'	'rpsslackg3_1106_monophase_lignec_court.mat'
'rpsslackg3_1077_tripahse_barreb4_long.mat'	'rpsslackg3_1107_monophase_ligned_long.mat'
'rpsslackg3_1078_tripahse_barreb4_court.mat'	'rpsslackg3_1108_monophase_ligned_court.mat'
'rpsslackg3_1079_tripahse_barreb5_long.mat'	'rpsslackg3_1109_monophase_lignef_long.mat'
'rpsslackg3_1080_tripahse_barreb5_court.mat'	'rpsslackg3_1110_monophase_lignef_court.mat'
'rpsslackg3_1081_biphase_lignea_long.mat'	'rpsslackg3_1111_monophase_barreb1_long.mat'
'rpsslackg3_1082_biphase_lignea_court.mat'	'rpsslackg3_1112_monophase_barreb1_court.mat'
'rpsslackg3_1083_biphase_ligne_b_long.mat'	'rpsslackg3_1113_monophase_barreb2_long.mat'
'rpsslackg3_1084_biphase_ligne_b_court.mat'	'rpsslackg3_1114_monophase_barreb2_court.mat'
'rpsslackg3_1085_biphase_lignec_long.mat'	'rpsslackg3_1115_monophase_barreb3_long.mat'
'rpsslackg3_1086_biphase_lignec_court.mat'	'rpsslackg3_1116_monophase_barreb3_court.mat'
'rpsslackg3_1087_biphase_ligned_long.mat'	'rpsslackg3_1117_monophase_barreb4_long.mat'
'rpsslackg3_1088_biphase_ligned_court.mat'	'rpsslackg3_1118_monophase_barreb4_court.mat'
'rpsslackg3_1089_biphase_lignef_long.mat'	'rpsslackg3_1119_monophase_barreb5_long.mat'
'rpsslackg3_1090_biphase_lignef_court.mat'	'rpsslackg3_1120_monophase_barreb5_court.mat'

Table 3.8 Tableau des simulations divergentes selon le COI pour le générateur 2



Générateur 2	
'rpsslackg4_1121_tripahse_lignea_long.mat'	'rpsslackg4_1151_biphase_barreb1_long.mat'
'rpsslackg4_1122_tripahse_lignea_court.mat'	'rpsslackg4_1152_biphase_barreb1_court.mat'
'rpsslackg4_1123_tripahse_ligne_b_long.mat'	'rpsslackg4_1153_biphase_barreb2_long.mat'
'rpsslackg4_1124_tripahse_ligne_b_court.mat'	'rpsslackg4_1154_biphase_barreb2_court.mat'
'rpsslackg4_1125_tripahse_lignec_long.mat'	'rpsslackg4_1155_biphase_barreb3_long.mat'
'rpsslackg4_1126_tripahse_lignec_court.mat'	'rpsslackg4_1156_biphase_barreb3_court.mat'
'rpsslackg4_1127_tripahse_ligned_long.mat'	'rpsslackg4_1157_biphase_barreb4_long.mat'
'rpsslackg4_1128_tripahse_ligned_court.mat'	'rpsslackg4_1158_biphase_barreb4_court.mat'
'rpsslackg4_1129_tripahse_lignef_long.mat'	'rpsslackg4_1159_biphase_barreb5_long.mat'
'rpsslackg4_1130_tripahse_lignef_court.mat'	'rpsslackg4_1160_biphase_barreb5_court.mat'
'rpsslackg4_1131_tripahse_barreb1_long.mat'	'rpsslackg4_1161_monophase_lignea_long.mat'
'rpsslackg4_1132_tripahse_barreb1_court.mat'	'rpsslackg4_1162_monophase_lignea_court.mat'
'rpsslackg4_1133_tripahse_barreb2_long.mat'	'rpsslackg4_1163_monophase_ligne_b_long.mat'
'rpsslackg4_1134_tripahse_barreb2_court.mat'	'rpsslackg4_1164_monophase_ligne_b_court.mat'
'rpsslackg4_1135_tripahse_barreb3_long.mat'	'rpsslackg4_1165_monophase_lignec_long.mat'
'rpsslackg4_1136_tripahse_barreb3_court.mat'	'rpsslackg4_1166_monophase_lignec_court.mat'
'rpsslackg4_1137_tripahse_barreb4_long.mat'	'rpsslackg4_1167_monophase_ligned_long.mat'
'rpsslackg4_1138_tripahse_barreb4_court.mat'	'rpsslackg4_1168_monophase_ligned_court.mat'
'rpsslackg4_1139_tripahse_barreb5_long.mat'	'rpsslackg4_1169_monophase_lignef_long.mat'
'rpsslackg4_1140_tripahse_barreb5_court.mat'	'rpsslackg4_1170_monophase_lignef_court.mat'
'rpsslackg4_1141_biphase_lignea_long.mat'	'rpsslackg4_1171_monophase_barreb1_long.mat'
'rpsslackg4_1142_biphase_lignea_court.mat'	'rpsslackg4_1172_monophase_barreb1_court.mat'
'rpsslackg4_1143_biphase_ligne_b_long.mat'	'rpsslackg4_1173_monophase_barreb2_long.mat'
'rpsslackg4_1144_biphase_ligne_b_court.mat'	'rpsslackg4_1174_monophase_barreb2_court.mat'
'rpsslackg4_1145_biphase_lignec_long.mat'	'rpsslackg4_1175_monophase_barreb3_long.mat'
'rpsslackg4_1146_biphase_lignec_court.mat'	'rpsslackg4_1176_monophase_barreb3_court.mat'
'rpsslackg4_1147_biphase_ligned_long.mat'	'rpsslackg4_1177_monophase_barreb4_long.mat'
'rpsslackg4_1148_biphase_ligned_court.mat'	'rpsslackg4_1178_monophase_barreb4_court.mat'
'rpsslackg4_1149_biphase_lignef_long.mat'	'rpsslackg4_1179_monophase_barreb5_long.mat'
'rpsslackg4_1150_biphase_lignef_court.mat'	'rpsslackg4_1180_monophase_barreb5_court.mat'

Table 3.9 Tableau des simulations divergentes selon le COI pour le générateur 2

Générateur 3	
'rpsslackg1_1001_tripahse_lignea_30_long.mat'	'rpsslackg1_1031_biphase_barreb1_long.mat'
'rpsslackg1_1002_tripahse_lignea_30_court.mat'	'rpsslackg1_1032_biphase_barreb1_court.mat'
'rpsslackg1_1003_tripahse_ligneb_30_long.mat'	'rpsslackg1_1033_biphase_barreb2_long.mat'
'rpsslackg1_1004_tripahse_ligneb_30_court.mat'	'rpsslackg1_1034_biphase_barreb2_court.mat'
'rpsslackg1_1005_tripahse_lignec_30_long.mat'	'rpsslackg1_1035_biphase_barreb3_long.mat'
'rpsslackg1_1006_tripahse_lignec_30_court.mat'	'rpsslackg1_1036_biphase_barreb3_court.mat'
'rpsslackg1_1007_tripahse_ligned_30_long.mat'	'rpsslackg1_1037_biphase_barreb4_long.mat'
'rpsslackg1_1008_tripahse_ligned_30_court.mat'	'rpsslackg1_1038_biphase_barreb4_court.mat'
'rpsslackg1_1009_tripahse_lignef_30_court.mat'	'rpsslackg1_1039_biphase_barreb5_long.mat'
'rpsslackg1_1010_tripahse_lignef_court.mat'	'rpsslackg1_1040_biphase_barreb5_court.mat'
	'rpsslackg1_1041_monophase_lignea_30_long.mat'
'rpsslackg1_1012_tripahse_barreb1_court.mat'	'rpsslackg1_1042_monophase_lignea_30_court.mat'
'rpsslackg1_1013_tripahse_barreb2_long.mat'	'rpsslackg1_1043_monophase_ligneb_30_long.mat'
'rpsslackg1_1014_tripahse_barreb2_court.mat'	'rpsslackg1_1044_monophase_ligneb_30_court.mat'
'rpsslackg1_1015_tripahse_barreb3_long.mat'	'rpsslackg1_1045_monophase_lignec_30_long.mat'
'rpsslackg1_1016_tripahse_barreb3_court.mat'	'rpsslackg1_1046_monophase_lignec_30_court.mat'
'rpsslackg1_1017_tripahse_barreb4_long.mat'	'rpsslackg1_1047_monophase_ligned_30_long.mat'
'rpsslackg1_1018_tripahse_barreb4_court.mat'	'rpsslackg1_1048_monophase_ligned_30_court.mat'
'rpsslackg1_1019_tripahse_barreb5_long.mat'	'rpsslackg1_1049_monophase_lignef_30_long.mat'
'rpsslackg1_1020_tripahse_barreb5_court.mat'	'rpsslackg1_1050_monophase_lignef_30_court.mat'
'rpsslackg1_1021_biphase_lignea_30_long.mat'	
'rpsslackg1_1022_biphase_lignea_30_court.mat'	'rpsslackg1_1052_monophase_barreb1_court.mat'
'rpsslackg1_1023_biphase_ligneb_30_long.mat'	'rpsslackg1_1053_monophase_barreb2_long.mat'
'rpsslackg1_1024_biphase_ligneb_30_court.mat'	'rpsslackg1_1054_monophase_barreb2_court.mat'
'rpsslackg1_1025_biphase_lignec_30_long.mat'	'rpsslackg1_1055_monophase_barreb3_long.mat'
'rpsslackg1_1026_biphase_lignec_30_court.mat'	'rpsslackg1_1056_monophase_barreb3_court.mat'
'rpsslackg1_1027_biphase_ligned_30_long.mat'	'rpsslackg1_1057_monophase_barreb4_court.mat'
'rpsslackg1_1028_biphase_ligned_30_court.mat'	'rpsslackg1_1058_monophase_barreb4_long.mat'
'rpsslackg1_1029_biphase_lignef_30_long.mat'	'rpsslackg1_1059_monophase_barreb5_long.mat'
'rpsslackg1_1030_biphase_lignef_30_court.mat'	'rpsslackg1_1060_monophase_barreb5_court.mat'

Table 3.10 Tableau des simulations divergentes selon le COI pour le générateur 3

Générateur 3	
'rpsslackg3_1061_tripahse_lignea_long.mat'	'rpsslackg3_1091_biphase_barreb1_long.mat'
'rpsslackg3_1062_tripahse_lignea_court.mat'	'rpsslackg3_1092_biphase_barreb1_court.mat'
'rpsslackg3_1063_tripahse_ligne_b_long.mat'	'rpsslackg3_1093_biphase_barreb2_long.mat'
'rpsslackg3_1064_tripahse_ligne_b_court.mat'	'rpsslackg3_1094_biphase_barreb2_court.mat'
'rpsslackg3_1065_tripahse_lignec_long.mat'	'rpsslackg3_1095_biphase_barreb3_long.mat'
'rpsslackg3_1066_tripahse_lignec_court.mat'	'rpsslackg3_1096_biphase_barreb3_court.mat'
'rpsslackg3_1067_tripahse_ligned_long.mat'	'rpsslackg3_1097_biphase_barreb4_long.mat'
'rpsslackg3_1068_tripahse_ligned_court.mat'	'rpsslackg3_1098_biphase_barreb4_court.mat'
'rpsslackg3_1069_tripahse_ligne_f_long.mat'	'rpsslackg3_1099_biphase_barreb5_long.mat'
'rpsslackg3_1070_tripahse_ligne_f_court.mat'	'rpsslackg3_1100_biphase_barreb5_court.mat'
'rpsslackg3_1071_tripahse_barreb1_long.mat'	'rpsslackg3_1101_monophase_lignea_long.mat'
'rpsslackg3_1072_tripahse_barreb1_court.mat'	'rpsslackg3_1102_monophase_lignea_court.mat'
'rpsslackg3_1073_tripahse_barreb2_long.mat'	'rpsslackg3_1103_monophase_ligne_b_long.mat'
'rpsslackg3_1074_tripahse_barreb2_court.mat'	'rpsslackg3_1104_monophase_ligne_b_court.mat'
'rpsslackg3_1075_tripahse_barreb3_long.mat'	'rpsslackg3_1105_monophase_lignec_long.mat'
'rpsslackg3_1076_tripahse_barreb3_court.mat'	'rpsslackg3_1106_monophase_lignec_court.mat'
'rpsslackg3_1077_tripahse_barreb4_long.mat'	'rpsslackg3_1107_monophase_ligned_long.mat'
'rpsslackg3_1078_tripahse_barreb4_court.mat'	'rpsslackg3_1108_monophase_ligned_court.mat'
'rpsslackg3_1079_tripahse_barreb5_long.mat'	'rpsslackg3_1109_monophase_ligne_f_long.mat'
'rpsslackg3_1080_tripahse_barreb5_court.mat'	'rpsslackg3_1110_monophase_ligne_f_court.mat'
'rpsslackg3_1081_biphase_lignea_long.mat'	'rpsslackg3_1111_monophase_barreb1_long.mat'
'rpsslackg3_1082_biphase_lignea_court.mat'	'rpsslackg3_1112_monophase_barreb1_court.mat'
'rpsslackg3_1083_biphase_ligne_b_long.mat'	'rpsslackg3_1113_monophase_barreb2_long.mat'
'rpsslackg3_1084_biphase_ligne_b_court.mat'	'rpsslackg3_1114_monophase_barreb2_court.mat'
'rpsslackg3_1085_biphase_lignec_long.mat'	'rpsslackg3_1115_monophase_barreb3_long.mat'
'rpsslackg3_1086_biphase_lignec_court.mat'	'rpsslackg3_1116_monophase_barreb3_court.mat'
'rpsslackg3_1087_biphase_ligned_long.mat'	'rpsslackg3_1117_monophase_barreb4_long.mat'
'rpsslackg3_1088_biphase_ligned_court.mat'	'rpsslackg3_1118_monophase_barreb4_court.mat'
'rpsslackg3_1089_biphase_ligne_f_long.mat'	'rpsslackg3_1119_monophase_barreb5_long.mat'
'rpsslackg3_1090_biphase_ligne_f_court.mat'	'rpsslackg3_1120_monophase_barreb5_court.mat'

Table 3.11 Tableau des simulations divergentes selon le COI pour le générateur 3

Générateur 3	
'rpsslackg4_1121_tripahse_lignea_long.mat'	'rpsslackg4_1151_biphase_barreb1_long.mat'
'rpsslackg4_1122_tripahse_lignea_court.mat'	'rpsslackg4_1152_biphase_barreb1_court.mat'
'rpsslackg4_1123_tripahse_ligne_b_long.mat'	'rpsslackg4_1153_biphase_barreb2_long.mat'
'rpsslackg4_1124_tripahse_ligne_b_court.mat'	'rpsslackg4_1154_biphase_barreb2_court.mat'
'rpsslackg4_1125_tripahse_lignec_long.mat'	'rpsslackg4_1155_biphase_barreb3_long.mat'
'rpsslackg4_1126_tripahse_lignec_court.mat'	'rpsslackg4_1156_biphase_barreb3_court.mat'
'rpsslackg4_1127_tripahse_ligned_long.mat'	'rpsslackg4_1157_biphase_barreb4_long.mat'
'rpsslackg4_1128_tripahse_ligned_court.mat'	'rpsslackg4_1158_biphase_barreb4_court.mat'
'rpsslackg4_1129_tripahse_ligne_f_long.mat'	'rpsslackg4_1159_biphase_barreb5_long.mat'
'rpsslackg4_1130_tripahse_ligne_f_court.mat'	'rpsslackg4_1160_biphase_barreb5_court.mat'
'rpsslackg4_1131_tripahse_barreb1_long.mat'	'rpsslackg4_1161_monophase_lignea_long.mat'
'rpsslackg4_1132_tripahse_barreb1_court.mat'	'rpsslackg4_1162_monophase_lignea_court.mat'
'rpsslackg4_1133_tripahse_barreb2_long.mat'	'rpsslackg4_1163_monophase_ligne_b_long.mat'
'rpsslackg4_1134_tripahse_barreb2_court.mat'	'rpsslackg4_1164_monophase_ligne_b_court.mat'
'rpsslackg4_1135_tripahse_barreb3_long.mat'	'rpsslackg4_1165_monophase_lignec_long.mat'
'rpsslackg4_1136_tripahse_barreb3_court.mat'	'rpsslackg4_1166_monophase_lignec_court.mat'
'rpsslackg4_1137_tripahse_barreb4_long.mat'	'rpsslackg4_1167_monophase_ligned_long.mat'
'rpsslackg4_1138_tripahse_barreb4_court.mat'	'rpsslackg4_1168_monophase_ligned_court.mat'
'rpsslackg4_1139_tripahse_barreb5_long.mat'	'rpsslackg4_1169_monophase_ligne_f_long.mat'
'rpsslackg4_1140_tripahse_barreb5_court.mat'	'rpsslackg4_1170_monophase_ligne_f_court.mat'
'rpsslackg4_1141_biphase_lignea_long.mat'	'rpsslackg4_1171_monophase_barreb1_long.mat'
'rpsslackg4_1142_biphase_lignea_court.mat'	'rpsslackg4_1172_monophase_barreb1_court.mat'
'rpsslackg4_1143_biphase_ligne_b_long.mat'	'rpsslackg4_1173_monophase_barreb2_long.mat'
'rpsslackg4_1144_biphase_ligne_b_court.mat'	'rpsslackg4_1174_monophase_barreb2_court.mat'
'rpsslackg4_1145_biphase_lignec_long.mat'	'rpsslackg4_1175_monophase_barreb3_long.mat'
'rpsslackg4_1146_biphase_lignec_court.mat'	'rpsslackg4_1176_monophase_barreb3_court.mat'
'rpsslackg4_1147_biphase_ligned_long.mat'	'rpsslackg4_1177_monophase_barreb4_long.mat'
'rpsslackg4_1148_biphase_ligned_court.mat'	'rpsslackg4_1178_monophase_barreb4_court.mat'
'rpsslackg4_1149_biphase_ligne_f_long.mat'	'rpsslackg4_1179_monophase_barreb5_long.mat'
'rpsslackg4_1150_biphase_ligne_f_court.mat'	'rpsslackg4_1180_monophase_barreb5_court.mat'

Table 3.12 Tableau des simulations divergentes selon le COI pour le générateur 3

Générateur 4	
'rpsslackg1_1001_tripahse_lignea_30_long.mat'	'rpsslackg1_1031_biphase_barreb1_long.mat'
'rpsslackg1_1002_tripahse_lignea_30_court.mat'	'rpsslackg1_1032_biphase_barreb1_court.mat'
'rpsslackg1_1003_tripahse_ligne_b_30_long.mat'	'rpsslackg1_1033_biphase_barreb2_long.mat'
'rpsslackg1_1004_tripahse_ligne_b_30_court.mat'	'rpsslackg1_1034_biphase_barreb2_court.mat'
'rpsslackg1_1005_tripahse_lignec_30_long.mat'	'rpsslackg1_1035_biphase_barreb3_long.mat'
'rpsslackg1_1006_tripahse_lignec_30_court.mat'	'rpsslackg1_1036_biphase_barreb3_court.mat'
'rpsslackg1_1007_tripahse_ligned_30_long.mat'	'rpsslackg1_1037_biphase_barreb4_long.mat'
'rpsslackg1_1008_tripahse_ligned_30_court.mat'	'rpsslackg1_1038_biphase_barreb4_court.mat'
'rpsslackg1_1009_tripahse_lignef_30_court.mat'	'rpsslackg1_1039_biphase_barreb5_long.mat'
'rpsslackg1_1010_tripahse_lignef_court.mat'	'rpsslackg1_1040_biphase_barreb5_court.mat'
	'rpsslackg1_1041_monophase_lignea_30_long.mat'
'rpsslackg1_1012_tripahse_barreb1_court.mat'	'rpsslackg1_1042_monophase_lignea_30_court.mat'
'rpsslackg1_1013_tripahse_barreb2_long.mat'	'rpsslackg1_1043_monophase_ligne_b_30_long.mat'
'rpsslackg1_1014_tripahse_barreb2_court.mat'	'rpsslackg1_1044_monophase_ligne_b_30_court.mat'
'rpsslackg1_1015_tripahse_barreb3_long.mat'	'rpsslackg1_1045_monophase_lignec_30_long.mat'
'rpsslackg1_1016_tripahse_barreb3_court.mat'	'rpsslackg1_1046_monophase_lignec_30_court.mat'
'rpsslackg1_1017_tripahse_barreb4_long.mat'	'rpsslackg1_1047_monophase_ligned_30_long.mat'
'rpsslackg1_1018_tripahse_barreb4_court.mat'	'rpsslackg1_1048_monophase_ligned_30_court.mat'
'rpsslackg1_1019_tripahse_barreb5_long.mat'	'rpsslackg1_1049_monophase_lignef_30_long.mat'
'rpsslackg1_1020_tripahse_barreb5_court.mat'	'rpsslackg1_1050_monophase_lignef_30_court.mat'
'rpsslackg1_1021_biphase_lignea_30_long.mat'	
'rpsslackg1_1022_biphase_lignea_30_court.mat'	'rpsslackg1_1052_monophase_barreb1_court.mat'
'rpsslackg1_1023_biphase_ligne_b_30_long.mat'	'rpsslackg1_1053_monophase_barreb2_long.mat'
'rpsslackg1_1024_biphase_ligne_b_30_court.mat'	'rpsslackg1_1054_monophase_barreb2_court.mat'
'rpsslackg1_1025_biphase_lignec_30_long.mat'	'rpsslackg1_1055_monophase_barreb3_long.mat'
'rpsslackg1_1026_biphase_lignec_30_court.mat'	'rpsslackg1_1056_monophase_barreb3_court.mat'
'rpsslackg1_1027_biphase_ligned_30_long.mat'	'rpsslackg1_1057_monophase_barreb4_court.mat'
'rpsslackg1_1028_biphase_ligned_30_court.mat'	'rpsslackg1_1058_monophase_barreb4_long.mat'
'rpsslackg1_1029_biphase_lignef_30_long.mat'	'rpsslackg1_1059_monophase_barreb5_long.mat'
'rpsslackg1_1030_biphase_lignef_30_court.mat'	'rpsslackg1_1060_monophase_barreb5_court.mat'

Table 3.13 Tableau des simulations divergentes selon le COI pour le générateur 4

Générateur 4	
'rpsslackg3_1061_tripahse_lignea_long.mat'	'rpsslackg3_1091_biphase_barreb1_long.mat'
'rpsslackg3_1062_tripahse_lignea_court.mat'	'rpsslackg3_1092_biphase_barreb1_court.mat'
'rpsslackg3_1063_tripahse_ligne_b_long.mat'	'rpsslackg3_1093_biphase_barreb2_long.mat'
'rpsslackg3_1064_tripahse_ligne_b_court.mat'	'rpsslackg3_1094_biphase_barreb2_court.mat'
'rpsslackg3_1065_tripahse_lignec_long.mat'	'rpsslackg3_1095_biphase_barreb3_long.mat'
'rpsslackg3_1066_tripahse_lignec_court.mat'	'rpsslackg3_1096_biphase_barreb3_court.mat'
'rpsslackg3_1067_tripahse_ligned_long.mat'	'rpsslackg3_1097_biphase_barreb4_long.mat'
'rpsslackg3_1068_tripahse_ligned_court.mat'	'rpsslackg3_1098_biphase_barreb4_court.mat'
'rpsslackg3_1069_tripahse_lignef_long.mat'	'rpsslackg3_1099_biphase_barreb5_long.mat'
'rpsslackg3_1070_tripahse_lignef_court.mat'	'rpsslackg3_1100_biphase_barreb5_court.mat'
'rpsslackg3_1071_tripahse_barreb1_long.mat'	'rpsslackg3_1101_monophase_lignea_long.mat'
'rpsslackg3_1072_tripahse_barreb1_court.mat'	'rpsslackg3_1102_monophase_lignea_court.mat'
'rpsslackg3_1073_tripahse_barreb2_long.mat'	'rpsslackg3_1103_monophase_ligne_b_long.mat'
'rpsslackg3_1074_tripahse_barreb2_court.mat'	'rpsslackg3_1104_monophase_ligne_b_court.mat'
'rpsslackg3_1075_tripahse_barreb3_long.mat'	'rpsslackg3_1105_monophase_lignec_long.mat'
'rpsslackg3_1076_tripahse_barreb3_court.mat'	'rpsslackg3_1106_monophase_lignec_court.mat'
'rpsslackg3_1077_tripahse_barreb4_long.mat'	'rpsslackg3_1107_monophase_ligned_long.mat'
'rpsslackg3_1078_tripahse_barreb4_court.mat'	'rpsslackg3_1108_monophase_ligned_court.mat'
'rpsslackg3_1079_tripahse_barreb5_long.mat'	'rpsslackg3_1109_monophase_lignef_long.mat'
'rpsslackg3_1080_tripahse_barreb5_court.mat'	'rpsslackg3_1110_monophase_lignef_court.mat'
'rpsslackg3_1081_biphase_lignea_long.mat'	'rpsslackg3_1111_monophase_barreb1_long.mat'
'rpsslackg3_1082_biphase_lignea_court.mat'	'rpsslackg3_1112_monophase_barreb1_court.mat'
'rpsslackg3_1083_biphase_ligne_b_long.mat'	'rpsslackg3_1113_monophase_barreb2_long.mat'
'rpsslackg3_1084_biphase_ligne_b_court.mat'	'rpsslackg3_1114_monophase_barreb2_court.mat'
'rpsslackg3_1085_biphase_lignec_long.mat'	'rpsslackg3_1115_monophase_barreb3_long.mat'
'rpsslackg3_1086_biphase_lignec_court.mat'	'rpsslackg3_1116_monophase_barreb3_court.mat'
'rpsslackg3_1087_biphase_ligned_long.mat'	'rpsslackg3_1117_monophase_barreb4_long.mat'
'rpsslackg3_1088_biphase_ligned_court.mat'	'rpsslackg3_1118_monophase_barreb4_court.mat'
'rpsslackg3_1089_biphase_lignef_long.mat'	'rpsslackg3_1119_monophase_barreb5_long.mat'
'rpsslackg3_1090_biphase_lignef_court.mat'	'rpsslackg3_1120_monophase_barreb5_court.mat'

Table 3.14 Tableau des simulations divergentes selon le COI pour le générateur 4

Générateur 4	
'rpsslackg4_1121_tripahse_lignea_long.mat'	'rpsslackg4_1151_biphase_barreb1_long.mat'
'rpsslackg4_1122_tripahse_lignea_court.mat'	'rpsslackg4_1152_biphase_barreb1_court.mat'
'rpsslackg4_1123_tripahse_ligne_b_long.mat'	'rpsslackg4_1153_biphase_barreb2_long.mat'
'rpsslackg4_1124_tripahse_ligne_b_court.mat'	'rpsslackg4_1154_biphase_barreb2_court.mat'
'rpsslackg4_1125_tripahse_lignec_long.mat'	'rpsslackg4_1155_biphase_barreb3_long.mat'
'rpsslackg4_1126_tripahse_lignec_court.mat'	'rpsslackg4_1156_biphase_barreb3_court.mat'
'rpsslackg4_1127_tripahse_ligned_long.mat'	'rpsslackg4_1157_biphase_barreb4_long.mat'
'rpsslackg4_1128_tripahse_ligned_court.mat'	'rpsslackg4_1158_biphase_barreb4_court.mat'
'rpsslackg4_1129_tripahse_lignef_long.mat'	'rpsslackg4_1159_biphase_barreb5_long.mat'
'rpsslackg4_1130_tripahse_lignef_court.mat'	'rpsslackg4_1160_biphase_barreb5_court.mat'
'rpsslackg4_1131_tripahse_barreb1_long.mat'	'rpsslackg4_1161_monophase_lignea_long.mat'
'rpsslackg4_1132_tripahse_barreb1_court.mat'	'rpsslackg4_1162_monophase_lignea_court.mat'
'rpsslackg4_1133_tripahse_barreb2_long.mat'	'rpsslackg4_1163_monophase_ligne_b_long.mat'
'rpsslackg4_1134_tripahse_barreb2_court.mat'	'rpsslackg4_1164_monophase_ligne_b_court.mat'
'rpsslackg4_1135_tripahse_barreb3_long.mat'	'rpsslackg4_1165_monophase_lignec_long.mat'
'rpsslackg4_1136_tripahse_barreb3_court.mat'	'rpsslackg4_1166_monophase_lignec_court.mat'
'rpsslackg4_1137_tripahse_barreb4_long.mat'	'rpsslackg4_1167_monophase_ligned_long.mat'
'rpsslackg4_1138_tripahse_barreb4_court.mat'	'rpsslackg4_1168_monophase_ligned_court.mat'
'rpsslackg4_1139_tripahse_barreb5_long.mat'	'rpsslackg4_1169_monophase_lignef_long.mat'
'rpsslackg4_1140_tripahse_barreb5_court.mat'	'rpsslackg4_1170_monophase_lignef_court.mat'
'rpsslackg4_1141_biphase_lignea_long.mat'	'rpsslackg4_1171_monophase_barreb1_long.mat'
'rpsslackg4_1142_biphase_lignea_court.mat'	'rpsslackg4_1172_monophase_barreb1_court.mat'
'rpsslackg4_1143_biphase_ligne_b_long.mat'	'rpsslackg4_1173_monophase_barreb2_long.mat'
'rpsslackg4_1144_biphase_ligne_b_court.mat'	'rpsslackg4_1174_monophase_barreb2_court.mat'
'rpsslackg4_1145_biphase_lignec_long.mat'	'rpsslackg4_1175_monophase_barreb3_long.mat'
'rpsslackg4_1146_biphase_lignec_court.mat'	'rpsslackg4_1176_monophase_barreb3_court.mat'
'rpsslackg4_1147_biphase_ligned_long.mat'	'rpsslackg4_1177_monophase_barreb4_long.mat'
'rpsslackg4_1148_biphase_ligned_court.mat'	'rpsslackg4_1178_monophase_barreb4_court.mat'
'rpsslackg4_1149_biphase_lignef_long.mat'	'rpsslackg4_1179_monophase_barreb5_long.mat'
'rpsslackg4_1150_biphase_lignef_court.mat'	'rpsslackg4_1180_monophase_barreb5_court.mat'

Table 3.15 Tableau des simulations divergentes selon le COI pour le générateur 4

3.3.5 Vecteur Status

Après avoir déterminé quelle simulation est stable et instable, nous ajoutons un vecteur *Status* à notre fichier de simulation. Celui-ci est initialisé à 0 depuis le début, jusqu'à la fin du court-circuit. À partir de là, chaque individu prend la valeur 0 si la simulation est stable ou 1 si la simulation est instable en regard des critères définis précédemment et ceci pour les 4 générateurs. Un exemple de vecteur *Status* présenté dans la figure suivante est tiré de la simulation 1052 :

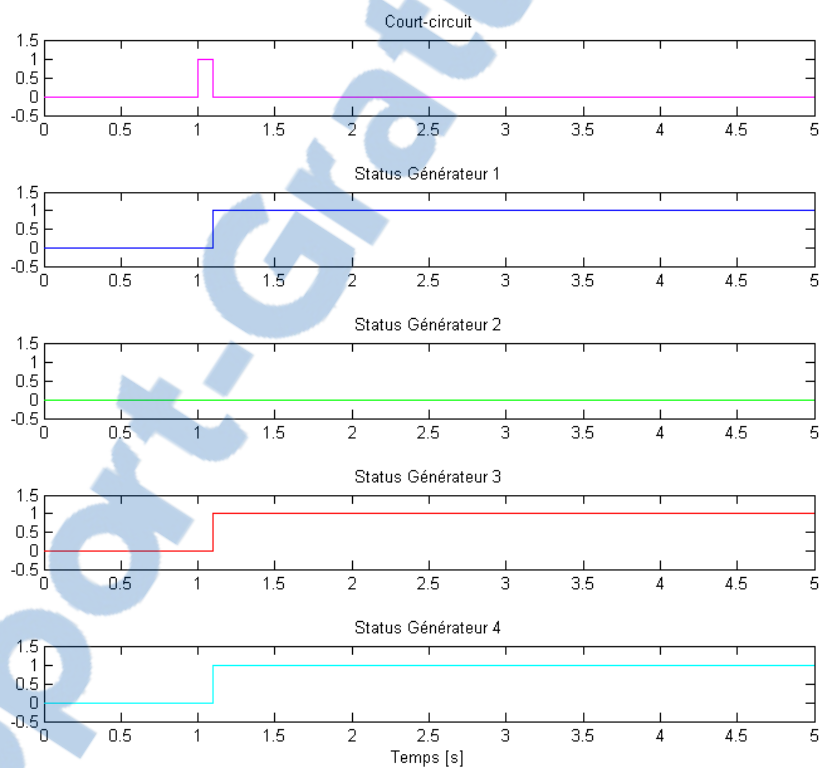


Figure 3.10 Vecteur status et court-circuit

3.4 Ajout de nouvelles variables

Afin d'améliorer les performances de prédiction du modèle, nous ajoutons d'autres variables qui peuvent être utiles à la prédiction de l'instabilité.

3.4.1 Phi et Cosinus Phi

Ayant déjà mesuré P_{RMS} et Q_{RMS} , nous pouvons calculer Phi, l'angle formé par la puissance active et la puissance apparente et ceci pour chaque phase. Nous ajoutons ces trois variables et par la suite, trois nouvelles autres variables sont créées en prenant le cosinus de ces dernières.

$$\varphi = \text{angle}(P_{RMS} + j \cdot Q_{RMS}) \quad (3.7)$$

3.4.2 SCV et dSCV/dt

Selon [5], le Swing-center voltage ou SCV est défini, dans un système à deux sources, comme l'endroit auquel la tension est nulle lorsque l'angle entre les deux sources E_S et E_R est de 180° . Lorsqu'un système est perturbé, l'angle δ va s'accroître et décroître. Si une instabilité en résulte, le vecteur SVC deviendra nul et le point o' se déplacera jusqu'à l'origine.

Afin d'avoir le SVC, nous devons connaître la tension interne et l'angle équivalent dans un système à deux sources, mais comme démontré dans [5], ce vecteur SCV peut être relativement bien approximé par l'équation suivante et étant donné que c'est sa dérivé qui nous donne la plus grande quantité d'information sur l'instabilité, une différence d'amplitude n'ont que peu d'impact dans le résultat final :

$$SCV \approx \|V_{S_{RMS}}\| \cdot \cos(\varphi) \quad (3.8)$$

Ou alors V_s est la tension de source et φ est la différence d'angle entre la tension et le courant :

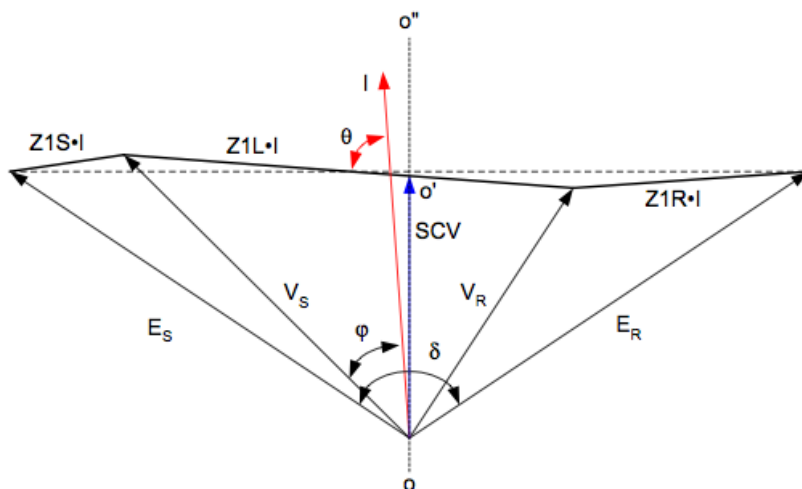


Figure 3.11 Illustration du SCV

Finalement, comme présenté dans le rapport [22], nous ajoutons le SCV ainsi que sa dérivée, mais aussi sa dérivée filtrée sur 1.5, 4 et 12 cycles, soit un total de quinze nouvelles variables.

3.4.3 Fréquence inter-barres et fréquence filtrée

Pour chaque point de mesure, nous mesurons la fréquence à cette barre et nous y soustrayons les autres fréquences des autres barres de mesures. Par exemple, pour le générateur 2, nous mesurons pour chaque phase :

- FreqGen2 - FreqGen1 ;
- FreqGen2 - FreqGen3 ;
- FreqGen2 - FreqGen4.

Finalement, comme présenté dans [22], nous prenons la dérivée filtrée sur trois et six cycles des fréquences de phases mesurées aux barres. Au final, ce sont quinze nouvelles autres variables qui sont ajoutées pour porter le total à 72 variables. Le tableau ci-dessous présente les variables avec leur numéro de colonne correspondant :



1	Phase_1/I_RMS	37	Id_T
2	Phase_1/I_T	38	Iq_RMS
3	Phase_1/P_RMS	39	Iq_T
4	Phase_1/P_T	40	V0_RMS
5	Phase_1/Q_RMS	41	V0_T
6	Phase_1/Q_T	42	Vd_RMS
7	Phase_1/V_RMS	43	Vd_T
8	Phase_1/V_T	44	Vq_RMS
9	Phase_2/I_RMS	45	Vq_T
10	Phase_2/I_T	46	Phase_1/UCosPhi
11	Phase_2/P_RMS	47	Phase_2/UCosPhi
12	Phase_2/P_T	48	Phase_3/UCosPhi
13	Phase_2/Q_RMS	49	Phase_1/dUdtUCosPhi/1.5Cycles
14	Phase_2/Q_T	50	Phase_2/dUdtUCosPhi/1.5Cycles
15	Phase_2/V_RMS	51	Phase_3/dUdtUCosPhi/1.5Cycles
16	Phase_2/V_T	52	Phase_1/dUdtUCosPhi/4Cycles
17	Phase_3/I_RMS	53	Phase_2/dUdtUCosPhi/4Cycles
18	Phase_3/I_T	54	Phase_3/dUdtUCosPhi/4Cycles
19	Phase_3/P_RMS	55	Phase_1/dUdtUCosPhi/12Cycles
20	Phase_3/P_T	56	Phase_2/dUdtUCosPhi/12Cycles
21	Phase_3/Q_RMS	57	Phase_3/dUdtUCosPhi/12Cycles
22	Phase_3/Q_T	58	Phase_1/Freq_Gen1_Gen2
23	Phase_3/V_RMS	59	Phase_2/Freq_Gen1_Gen2
24	Phase_3/V_T	60	Phase_3/Freq_Gen1_Gen2
25	DEV4/Phi	61	Phase_1/Freq_Gen1_Gen3
26	DEV6/Phi	62	Phase_2/Freq_Gen1_Gen3
27	DEV8/Phi	63	Phase_3/Freq_Gen1_Gen3
28	Cos_Phi_1	64	Phase_1/Freq_Gen1_Gen4
29	Cos_Phi_2	65	Phase_2/Freq_Gen1_Gen4

30	Cos_Phi_3	66	Phase_3/Freq_Gen1_Gen4
31	Freq_v1T	67	Phase_1/Freq_Filt_3cy
32	Freq_v2T	68	Phase_2/Freq_Filt_3cy
33	Freq_v3T	69	Phase_3/Freq_Filt_3cy
34	I0_RMS	70	Phase_1/Freq_Filt_6cy
35	I0_T	71	Phase_2/Freq_Filt_6cy
36	Id_RMS	72	Phase_3/Freq_Filt_6cy

Table 3.16 Variables prédictives et le numéro de colonne

3.5 Création des différents ensembles

Afin d'entraîner un modèle prédictif, le document [11] veut idéalement que 60% de tout l'ensemble d'individus soient dédiés pour l'apprentissage du modèle de prédiction et le reste doit être dédié au test et à la validation du modèle.

La méthode d'échantillonnage de simulation est utilisée dans notre cas plutôt que les individus afin de gagner en lisibilité et traçabilité. En effet, si nous avons échantillonné les données, il serait difficile de savoir quelle donnée provient de quelle simulation et par conséquent il serait difficile de découvrir des faiblesses de notre modèle de prédiction vis-à-vis de certaines simulations. De plus, nous effectuons un échantillonnage stratifié afin de préserver le ratio de simulations stables et instables et ainsi éviter qu'un ensemble ne se retrouve sans individus stables ou instables.

Pour notre cas, 60% des simulations seront utilisées pour l'apprentissage, 15% seront utilisées pour le test et 25% pour la validation du modèle. Les tableaux ci-dessous résument les simulations qui sont affectées aux différents jeux de données.

Au total, comme calculé au paragraphe 3.3.3, nous avons :

	Générateur 1	Générateur 2	Générateur 3	Générateur 4
Stables	152	143	136	151
Instables	26	35	42	27

Table 3.17 Répartition des simulations au total

Le 60% des simulations est dédié à l'apprentissage soit :

	Générateur 1	Générateur 2	Générateur 3	Générateur 4
Stables	91	86	82	91
Instables	16	21	25	16

Table 3.18 Nombre de simulations dédiées à l'apprentissage

Le 15% des simulations dédiées au test soit :

	Générateur 1	Générateur 2	Générateur 3	Générateur 4
Stables	23	21	20	23
Instables	4	5	6	4

Table 3.19 Nombre de simulations dédiées au test

Et le reste à l'ensemble de validation soit :

	Générateur 1	Générateur 2	Générateur 3	Générateur 4
Stables	38	36	34	37
Instables	6	9	11	7

Table 3.20 Nombre de simulations dédiées à la validation

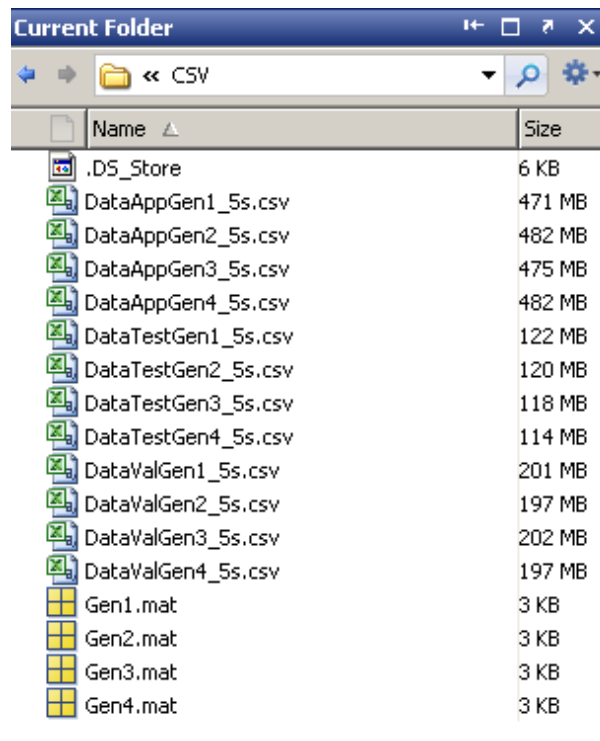
L'annexe 3 montre quelle simulation est affectée à quoi pour les 4 générateurs. Étant donné que nous avons choisi de travailler un modèle de prédiction pour chaque générateur, il est possible que pour la même simulation, les individus des générateurs 1 servent à l'apprentissage et les mesures du générateur 2 servent à la validation par exemple.

3.6 Exportation des données

Afin de pouvoir utiliser KNIME pour la modélisation du modèle de prédiction, les données sont exportées dans un format .CSV avec leur entête respective en utilisant le script créé par [23]. Au préalable, nous avons retiré toutes les données jusqu'à l'interruption du court-circuit et nous n'avons gardé que les cinq secondes après la contingence car la prédiction de l'état du système se fera après que le court-circuit ne soit arrêté.

Avant l'exportation en format .CSV, la fréquence d'échantillonnage a dû être réduite à 1920 [Hz] car, lors de la concaténation des individus des différentes simulations, la mémoire de l'ordinateur était saturée.

Au final, douze fichiers sont exportés et quatre matrices MATLAB contiennent les simulations utilisés et un pointeur servant à identifier le début d'une nouvelle simulation:



Name	Size
.DS_Store	6 KB
DataAppGen1_5s.csv	471 MB
DataAppGen2_5s.csv	482 MB
DataAppGen3_5s.csv	475 MB
DataAppGen4_5s.csv	482 MB
DataTestGen1_5s.csv	122 MB
DataTestGen2_5s.csv	120 MB
DataTestGen3_5s.csv	118 MB
DataTestGen4_5s.csv	114 MB
DataValGen1_5s.csv	201 MB
DataValGen2_5s.csv	197 MB
DataValGen3_5s.csv	202 MB
DataValGen4_5s.csv	197 MB
Gen1.mat	3 KB
Gen2.mat	3 KB
Gen3.mat	3 KB
Gen4.mat	3 KB

Figure 3.12 Espace disque utilisé par les fichiers exportés

4. Modélisation et évaluation du prédicteur

En premier lieu, dans ce chapitre, nous expliquons notre choix pour le logiciel KNIME et nous présenterons brièvement celui-ci.

Par la suite, nous présenterons la qualité de prédiction de chacun des différents prédicteurs en fonction de leur fréquence d'échantillonnage et un résumé sera fait pour savoir quel prédicteur choisir à quelle fréquence d'échantillonnage.

La section d'après traitera de la réduction des variables, nous évaluerons chaque méthode de filtrage avec leurs algorithmes de recherche correspondant et nous résumerons les différentes méthodes.

Finalement, la dernière section utilisera l'ensemble de validation et présentera comment le prédicteur réagit face à celui-ci.

4.1 KNIME

Une quantité de programmes de fouille de données statistiques est disponible. Parmi eux, nous pouvons citer les versions à licence payante :

- MATLAB ;
- Statistica Data Miner ;
- KXEN Analytic Framework ;
- SPSS ;
- SAS Entreprise Miner.

Et les logiciels à licence gratuite tels que :

- Weka ;
- RapidMiner ;
- Rattle et R ;
- Tanagra ;
- Orange ;
- Spina ;
- KNIME.

KNIME, développé par l'université de Constance en Allemagne, est choisi car il permet de swapper les données sur le disque dur et ainsi de lever la contrainte de la mémoire vive [24] et ainsi de pouvoir manipuler une quantité de données plus importante. La prise en main du logiciel est facile, conviviale et intuitive grâce à son interface graphique et ses capacités peuvent être étendues car les bibliothèques WEKA et R lui offrent une grande flexibilité. La figure ci-dessous montre un workflow simple :

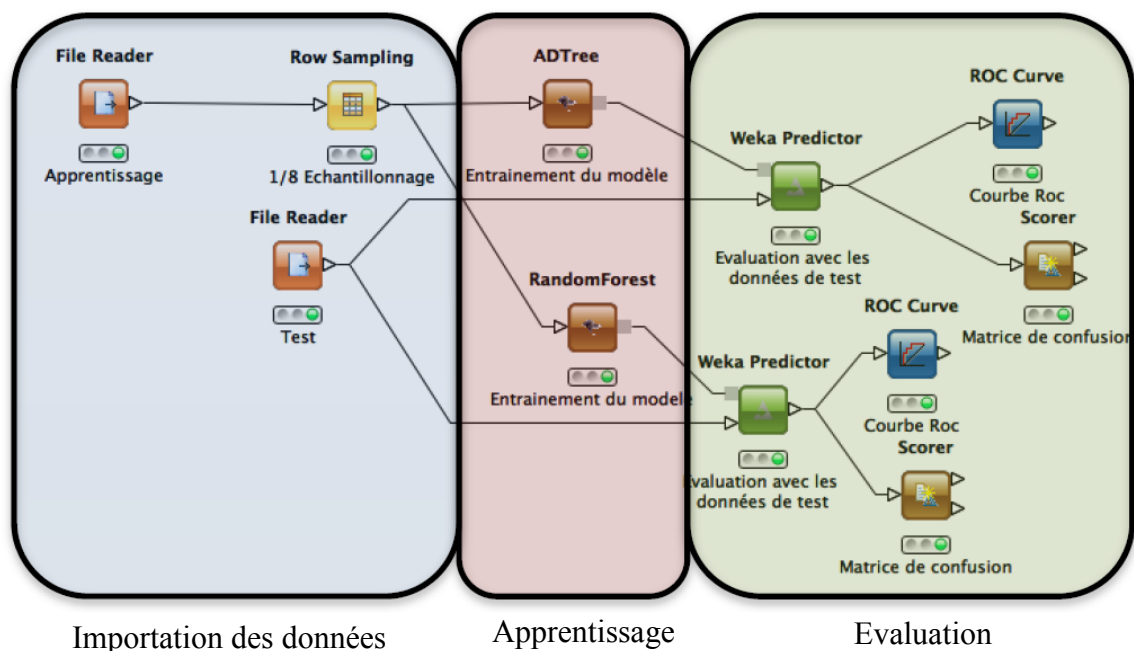


Figure 4.1 Workflow en utilisant KNIME

4.2 Baisse de la fréquence d'échantillonnage

La fréquence d'échantillonnage est de 1920 [Hz], soit 72 variables prédictive et 1920 individus collectés par seconde. Ce sous-chapitre a pour but de vérifier la pertinence de cette vitesse d'échantillonnage et de voir l'effet sur les performances de prédiction de plusieurs modèles.

Pour ce faire, en partant de la fréquence d'échantillonnage nominale, nous divisons à chaque fois la fréquence d'échantillonnage par deux, ce qui revient à prendre un individu sur deux, puis un individu sur quatre etc. Les résultats seront analysés au chapitre 4.2.9.

4.2.1 C4.5 Pruned

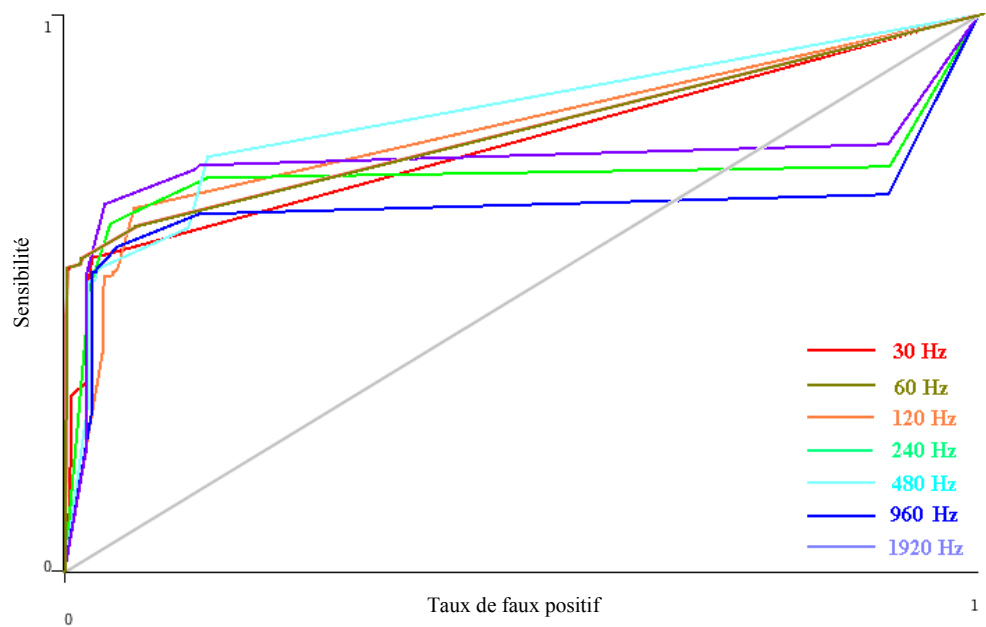


Figure 4.2 Courbes ROC C4.5 Pruned

Freq [Hz]	30	60	120	240	480	960	1920
AUC	0.7673	0.7919	0.7912	0.7108	0.8178	0.6551	0.7391

Table 4.1 Mesures AUC en fonction de la fréquence C4.5 Pruned

4.2.2 C4.5 Unpruned

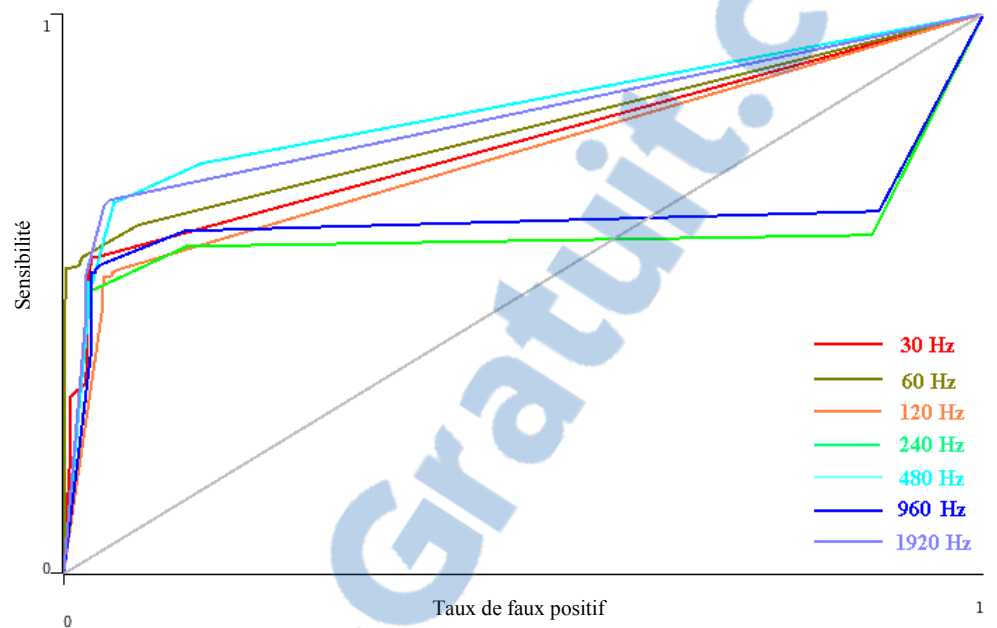


Figure 4.3 Courbes ROC C4.5 Unpruned

Freq [Hz]	30	60	120	240	480	960	1920
AUC	0.7673	0.7914	0.7447	0.6061	0.8257	0.6348	0.813

Table 4.2 Mesures AUC en fonction de la fréquence C4.5 Unpruned

4.2.3 CART Pruned

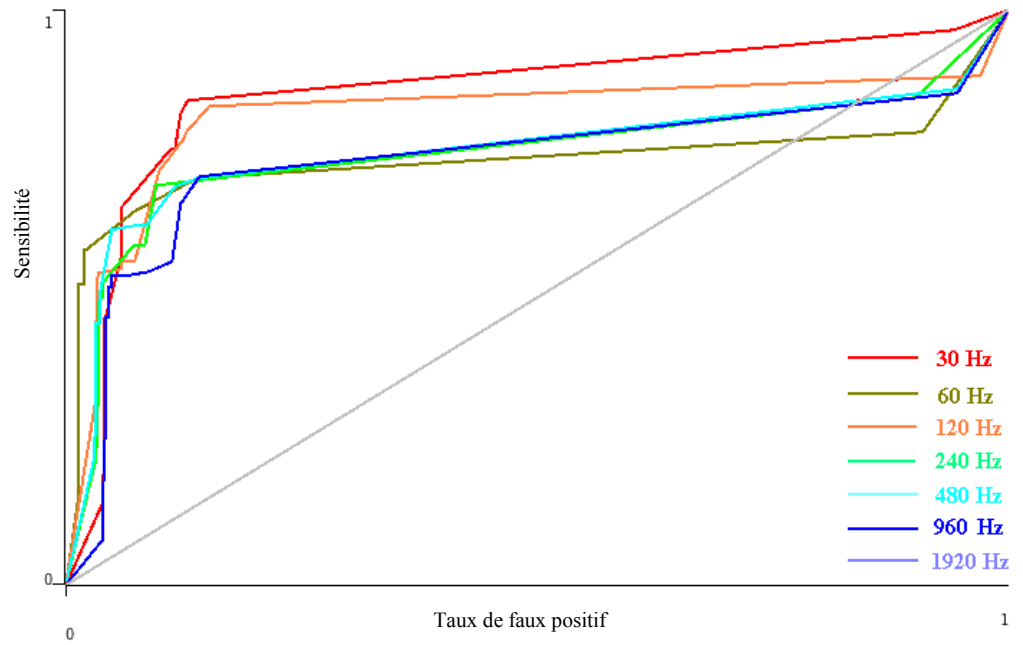


Figure 4.4 Courbes ROC CART Pruned

Freq [Hz]	30	60	120	240	480	960	1920
AUC	0.8564	0.739	0.8163	0.7519	0.7558	0.7339	0.6585

Table 4.3 Mesures AUC en fonction de la fréquence CART Pruned

4.2.4 CART Unpruned

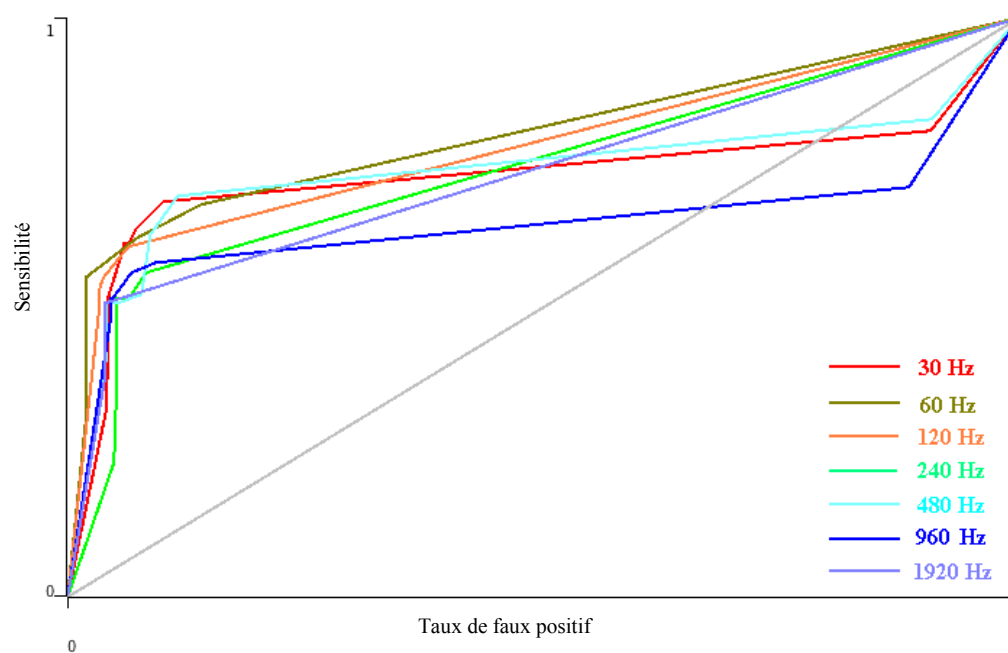


Figure 4.5 Courbes ROC CART Unpruned

Freq [Hz]	30	60	120	240	480	960	1920
AUC	0.7277	0.7997	0.7772	0.7381	0.7368	0.6449	0.7317

Table 4.4 Mesures AUC en fonction de la fréquence CART Unpruned



4.2.5 ADABOOST

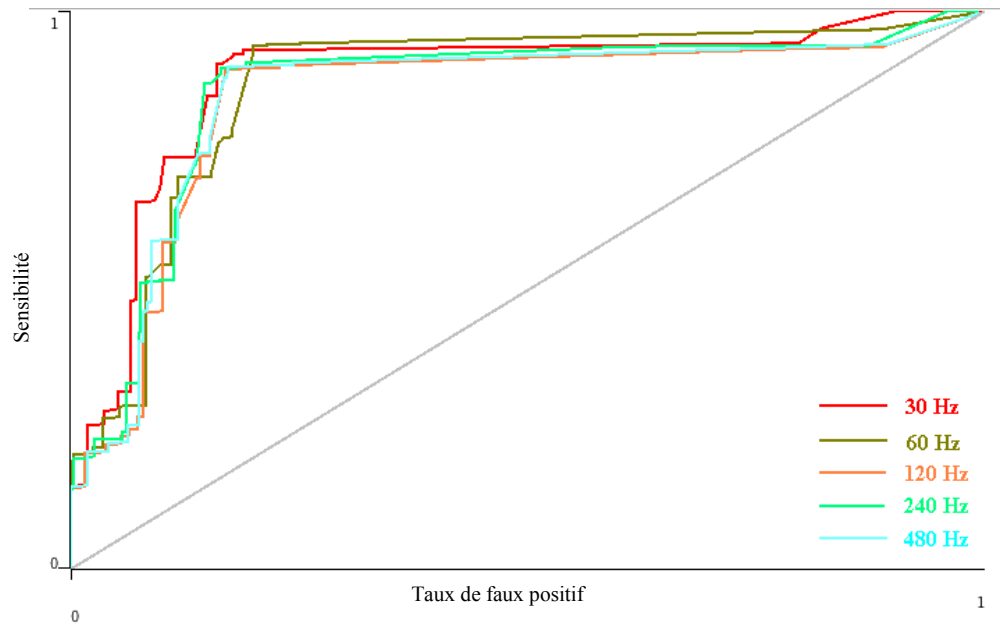


Figure 4.6 Courbes ROC ADABOOST

Freq [Hz]	30	60	120	240	480
AUC	0.8812	0.8704	0.8436	0.8436	0.8493

Table 4.5 Mesures AUC en fonction de la fréquence ADABOOST

4.2.6 ADTree

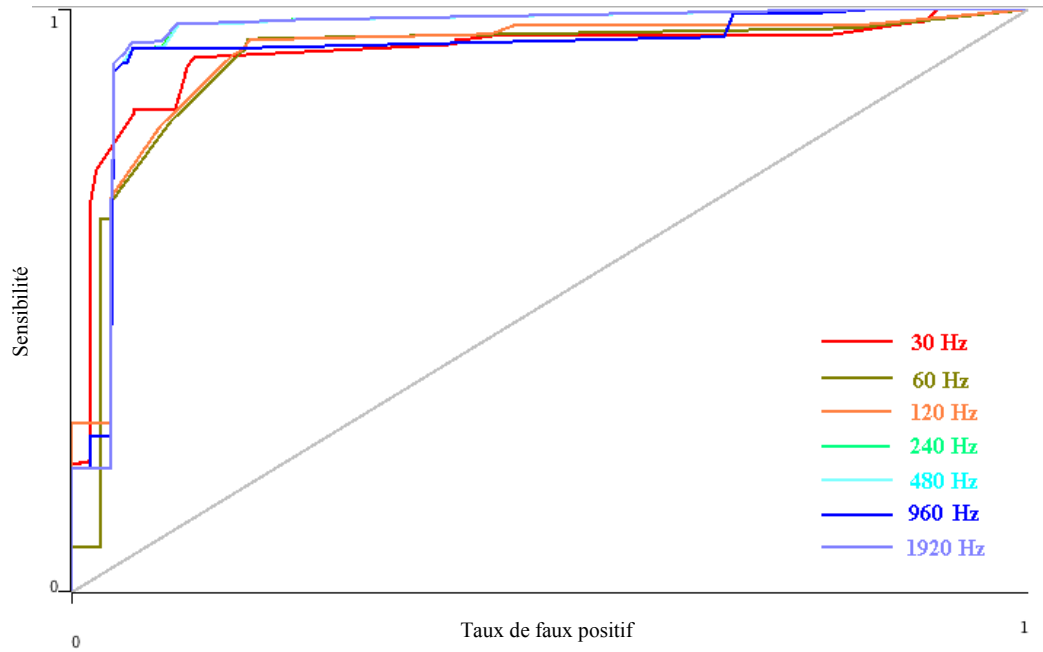


Figure 4.7 Courbes ROC ADTree

Freq [Hz]	30	60	120	240	480	960	1920
AUC	0.9239	0.9134	0.9212	0.9536	0.9536	0.9285	0.9551

Table 4.6 Mesures AUC en fonction de la fréquence ADTree

4.2.7 10 forêts aléatoires

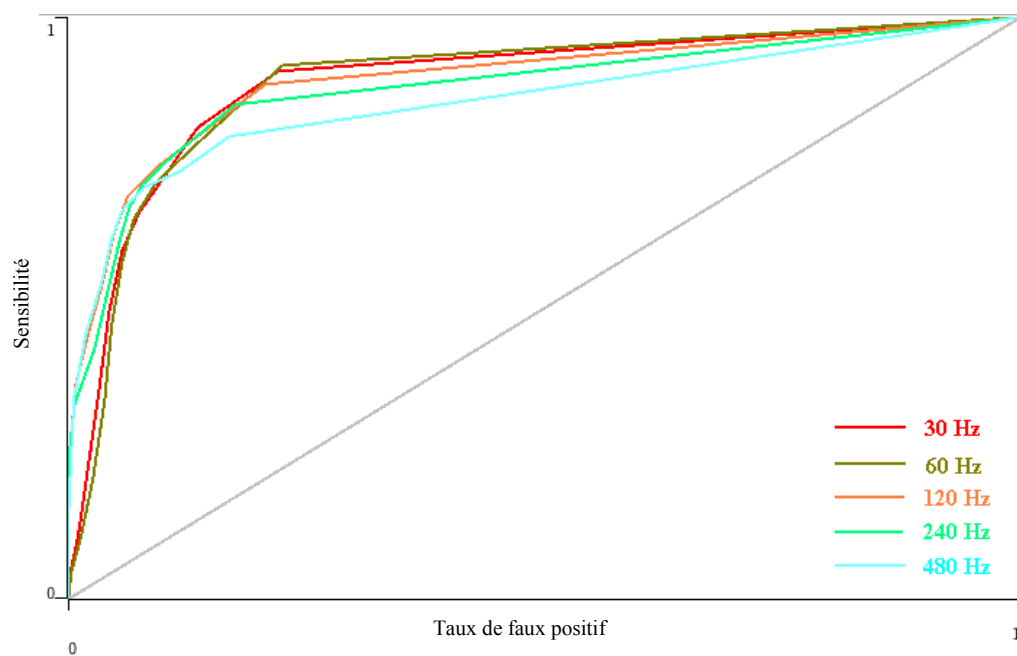


Figure 4.8 Courbes ROC 10 forêts aléatoires

Freq [Hz]	30	60	120	240	480
AUC	0.8906	0.8899	0.8939	0.8791	0.8569

Table 4.7 Mesures AUC en fonction de la fréquence 10 forêts aléatoires

4.2.8 100 forêts aléatoires

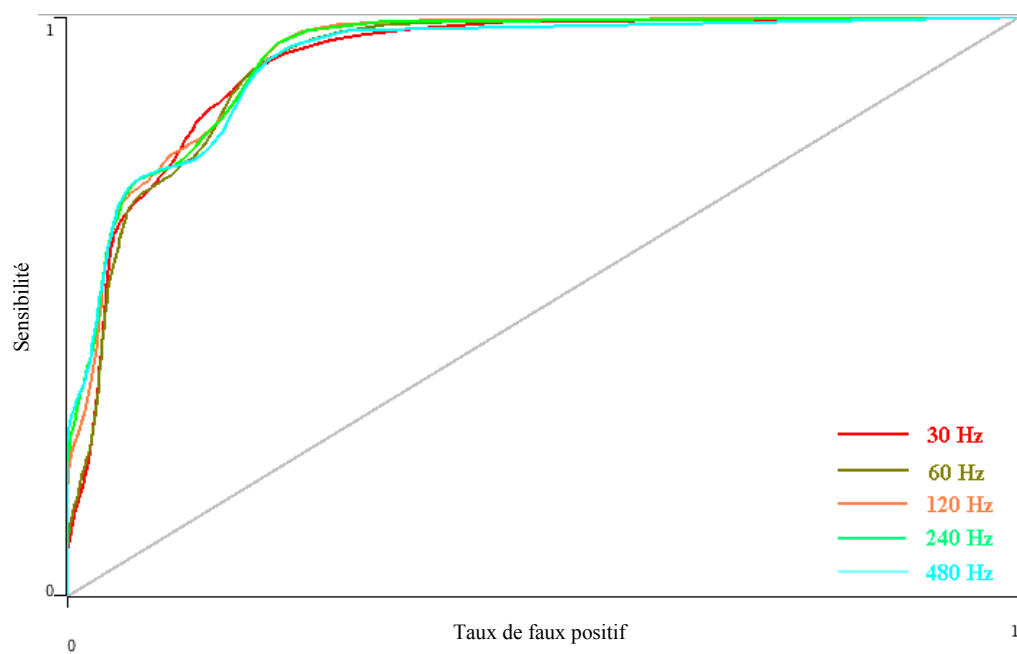


Figure 4.9 Courbes ROC 100 forêts aléatoires

Freq [Hz]	30	60	120	240	480
AUC	0.9228	0.9232	0.9317	0.9328	0.9249

Table 4.8 Mesures AUC en fonction de la fréquence 100 forêts aléatoires

4.2.9 Résumé et analyse

Le tableau et la figure ci-dessous résument les résultats obtenus pour le générateur 1.

$\text{Log}_2(\text{freq})$	Freq	%	C4.5 P	C4.5 UP	Cart P	Cart UP	ADTree	Adaboost	RF10	RF100
4.91	30	1.5625	0.7673	0.7673	0.8564	0.7277	0.9239	0.8812	0.8906	0.9228
5.91	60	3.125	0.7919	0.7914	0.739	0.7997	0.9134	0.8704	0.8899	0.9232
6.91	120	6.25	0.7912	0.7447	0.8163	0.7772	0.9212	0.8436	0.8939	0.9317
7.91	240	12.5	0.7108	0.6061	0.7519	0.7381	0.9536	0.8436	0.8791	0.9328
8.91	480	25	0.8178	0.8257	0.7558	0.7368	0.9536	0.8493	0.8569	0.9249
9.91	960	50	0.6551	0.6348	0.7339	0.6449	0.9285			
10.91	1920	100	0.7391	0.813	0.6585	0.7317	0.9551			

Table 4.9 Mesure AUC pour le générateur 1

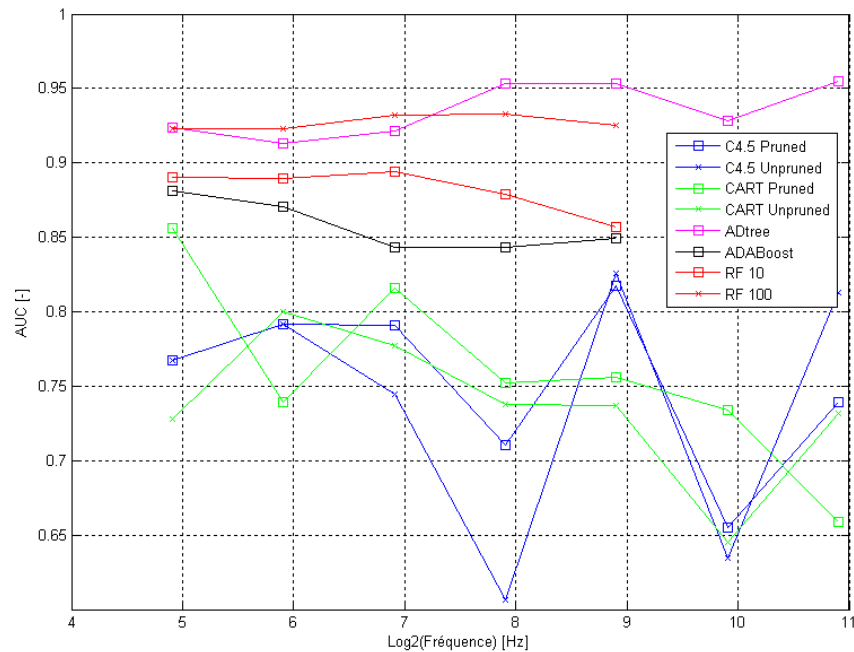


Figure 4.10 AUC en fonction du $\log_2(\text{fréquence})$

Nous voyons d'abord que l'échantillonnage a un grand impact sur la qualité de la prédiction obtenue par les arbres de décision classiques.

Ceci est dû à la variation de la variance causée par l'échantillonnage. Nous voyons aussi que l'élagage de l'arbre classique améliore les prédictions ou les réduit dépendamment de la fréquence d'échantillonnage.

Nous constatons que les techniques de décisions améliorées tel que ADTree, ADABOOST ou les forêts aléatoires offrent d'une part une meilleure et plus constante performance sur toute la plage d'échantillonnage.

Nous voyons qu'augmenter le nombre d'arbres dans la forêt aléatoire réduit la variance des individus (constante AUC plus constante) et améliore la prédiction. D'un point de vue statistique, plus il y a d'arbres de décisions dans une forêt aléatoire, mieux c'est. Dans la pratique, nous sommes, comme pour la technique ADABOOST, limités par les capacités informatiques et c'est pourquoi les coefficients AUC sont manquants pour 50% et 100%.

Les meilleures prédictions sont obtenues par l'arbre ADTree. Nous voyons aussi une chute des performances à 960 [Hz] et en-dessous de 240 [Hz]. À la vue de ce graphique, deux décisions peuvent être prises.

- Une fréquence d'échantillonnage de 240 [Hz] offre un bon compromis entre qualité de prédiction et ressources informatiques.
- Les arbres de décision classiques ainsi que les arbres de décision ADABOOST offrent une moins bonne qualité de prédiction que les forêts aléatoires et les arbres de type ADTree et ne seront pas utilisés pour la suite.

4.3 Réduction du nombre de variables

Pour les quatre générateurs, à l'aide des différents algorithmes décrits dans le chapitre 1, nous tenterons de réduire la quantité de prédicteurs nécessaires et nous en analyserons les performances. En utilisant une fréquence d'échantillonnage de 240 [Hz], nous utiliserons l'algorithme. Nous prendrons en référence la Table 3.16

Le tableau ci-dessous nous montre visuellement les prédicteurs choisis. Les colonnes représentent les générateurs alors que les lignes représentent les variables. Si l'élément du tableau est coloré, c'est que la variable est pertinente et doit être gardée.

Nous constatons qu'avec cette méthode, les générateurs 1 et 4 utilisent moins de variables prédictives et que les variables des fréquences, soit les colonnes 58 et plus, sont très sollicitées.

4.3.2 CFS - Rank Search

Générateur 1 : 10, 64, 65, 66, 68, soit 5 variables.

Générateur 2 : 9, 11, 17, 31, 32, 33, 34, 37, 38, 40, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, soit 33 variables.

Générateur 3 : 9, 10, 12, 13, 15, 17, 21, 31, 32, 33, 34, 35, 38, 39, 40, 41, 43, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, soit 41 variables.

Générateur 4 : 7, 15, 23, 26, 29, 31, 33, 34, 40, 46, 48, 57, 58, 59, 60, soit 15 variables.

Comme dans le cas précédent, nous voyons que les générateurs 1 et 4 ont besoin de moins de variables et que les variables de fréquence sont utilisées, surtout pour les petits générateurs.

Comme dans la majorité des cas vus jusqu'à présent, les deux gros générateurs 1 et 4 ont besoin de moins de variables prédictives et comme précédemment, nous voyons que c'est les variables contenant des mesures de fréquence qui sont les plus sollicitées. Nous voyons aussi qu'au milieu du tableau, les mesures 31, 32 et 33 sont sélectionnées et elles correspondent à la mesure de fréquence de barre.

4.3.6 Résumé du nombre de variables sélectionnées

Le tableau ci-dessous résume le nombre de variables sélectionnées pour chaque méthode

	CFS BF	CFS RS	SYMM FCBF	Const BF	Const RS
Générateur 1	7	5	2	6	17
Générateur 2	22	33	1	8	26
Générateur 3	26	41	5	9	20
Générateur 4	5	15	2	5	9
Total	60	94	10	28	72

Table 4.14 Nombre de variables sélectionnées par filtrage

Nous voyons que la méthode Symmetrical Uncert Attribute Set est celle qui sélectionne le moins de variables. Nous voyons que dans la majorité des cas, pour les générateurs 1 et 4, moins de variables sont sélectionnées.

4.3.7 Performances de prédiction

Après avoir sélectionné les forêts aléatoires et les arbres ADTree, nous évaluons les performances de ces prédicteurs avec les variables réduites et nous les comparerons avec les jeux de variables tout entiers.

En utilisant comme modèle de prédiction les forêts aléatoires nous obtenons les résultats suivants :

	CFS BF	CFS RS	Symmetry FCBF	Consistency BF	Consistency RS	72 Variables
Gen 1	0.8769	0.881	0.8202	0.9173	0.9196	0.9328
Gen 2	0.8711	0.8667	0.6832	0.8659	0.8659	0.9033
Gen 3	0.9683	0.9688	0.906	0.9791	0.9818	0.9447
Gen 4	0.9573	0.9475	0.835	0.9573	0.9474	0.9639
Moyenne	0.918	0.916	0.811	0.9299	0.928	0.9361

Table 4.15 AUC des 4 générateurs avec filtrage de variable et forêts aléatoires

Et pour l'arbre ADTree, nous obtenons :

	CFS BF	CFS RS	Symmetry FCBF	Consistency BF	Consistency RS	72 Variables
Gen 1	0.8958	0.8701	0.8586	0.9145	0.9443	0.9536
Gen 2	0.8238	0.8448	0.7819	0.8791	0.8793	0.8747
Gen 3	0.9042	0.9097	0.8826	0.9448	0.8286	0.9334
Gen 4	0.9117	0.938	0.8824	0.9117	0.9275	0.9435
Moyenne	0.8838	0.890	0.851	0.9125	0.8945	0.9263

Table 4.16 AUC des 4 générateurs avec filtrage de variable et ADtree

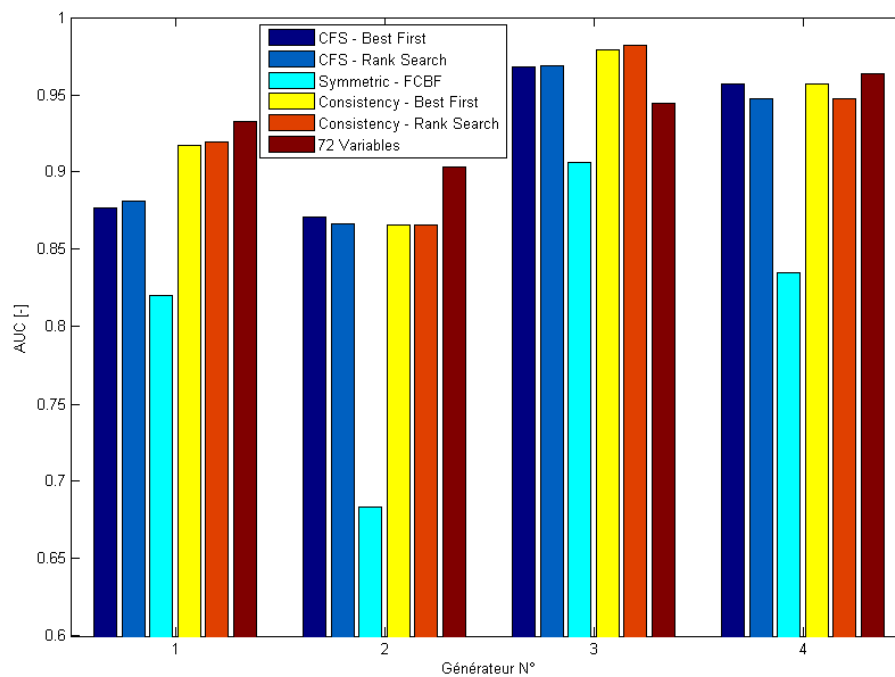


Figure 4.11 Comparaison de performances des réducteurs de variables - Forêts aléatoires

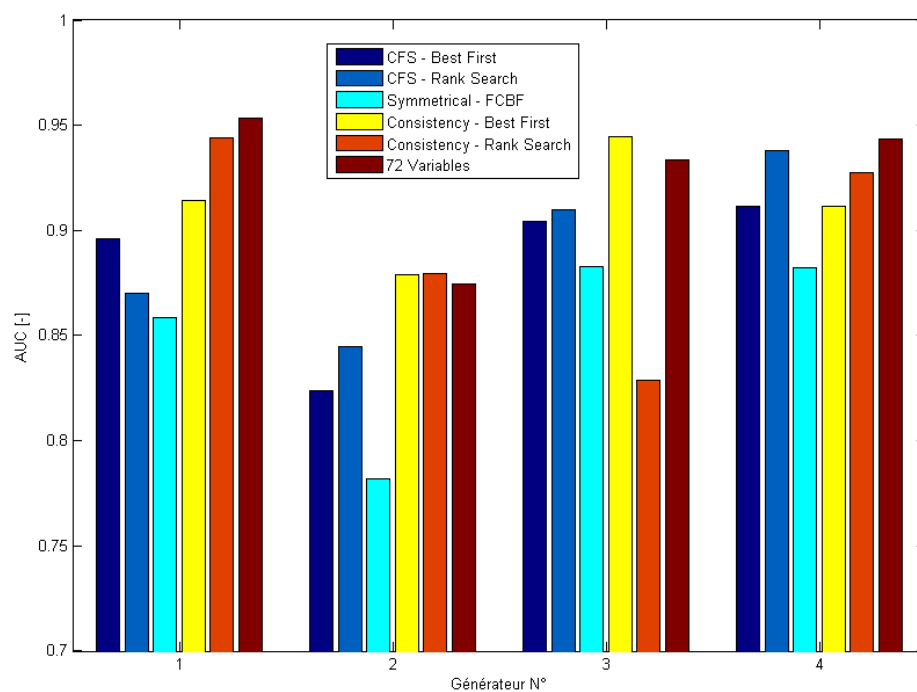


Figure 4.12 Comparaison de performances des réducteurs de variables - ADTree

A l'aide des graphiques et des tableaux précédents, nous concluons que :

- la méthode Symmetrical Uncert Attribute Set est celle qui donne les moins bons résultats ;
- la méthode de filtrage Consistency Subset Eval est la meilleure méthode de filtrage car, en moyenne, c'est celle qui offre l'aire sous la courbe la plus élevée ;
- Les méthodes de filtrage réduisent les performance en comparaison à la modélisation avec tous le jeux de variables ;
- Les algorithmes de recherche Best First et Rank Search offrent des performances comparables ;
- Les forêts aléatoires offrent de meilleurs performances que les ADTrees mais ces derniers sont plus rapides à l'apprentissage du modèle.

Comme nous l'avons vu dans les sous-chapitres précédent les méthodes Consistency Subset Eval réduisent le nombre de variables prédictives drastiquement. En effet, nous passons de 288 variables pour les 4 générateurs à seulement 28 en utilisant l'algorithme de recherche Best First et à 72 en utilisant Rank Search.

Finalement, en choisissant les forêts aléatoires et la méthode de filtrage Consistency Subset Eval - Best First, nous supprimons 90% des variables mais nous perdons que 0.6% de qualité de prédiction. Le nom des variables sélectionné pour chaque générateur est présenté dans le tableau suivant :

Générateur 1	Phase_1/I_RMS Phase_1/P_T Phase_3/Q_RMS I0_RMS Phase_3/Freq__Gen4 Phase_2/Freq_Filt_6cy	Générateur 2	Phase_1/P_RMS Phase_3/I_T Phase_3/V_RMS I0_RMS Id_T Phase_2/Freq_Gen2_Gen4 Phase_3/Freq_Gen2_Gen4 Phase_1/Freq_Filt_3cy
Générateur 3	Phase_1/I_T Phase_2/P_RMS I0_RMS Iq_T V0_RMS Vd_T Phase_2/dUdtUCosPhi4Cycles Phase_1/Freq_Gen3_Gen1 Phase_1/Freq_Gen3_Gen2	Générateur 4	Phase_1/I_RMS I0_RMS Vq_T Phase_3Freq_Gen4_Gen1 Phase_2Freq_Filt_6cy

Table 4.17 Résumé des variables sélectionnées pour chaque générateur

4.4 Validation

Après avoir modélisé et testé les 100 forêts aléatoires avec la réduction de variables, nous utilisons l'ensemble de validation afin de voir la qualité de la prédiction sur les quatre générateurs. Le load flow correspondant sur KNIME est le suivant :

Nous exportons les variables Status et Winner qui correspondent à l'état réel et celui prédit afin de les comparer visuellement grâce à MATLAB.

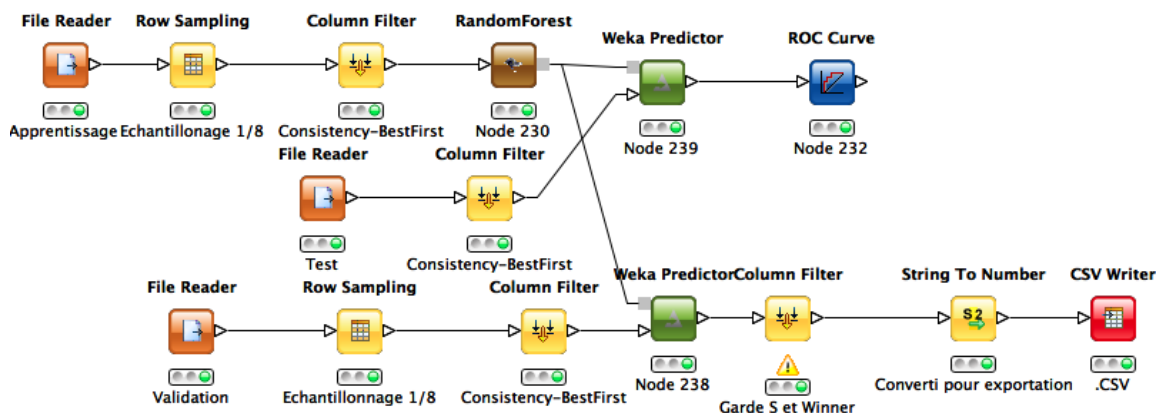


Figure 4.13 Work flow KNIME pour l'ensemble de validation

Pour les quatre générateurs, toutes les simulations sont mises bout à bout. En bleu, il s'agit du vecteur *Status* déterminé par le COIA et COIQ. Celui-ci a un offset de 0.1 afin de mieux le voir. En vert, il s'agit du vecteur *Winner*, soit celui prédit par les forêts aléatoires. Finalement, les carrés rouges indiquent le début d'une nouvelle simulation.

4.4.1 Générateur 1

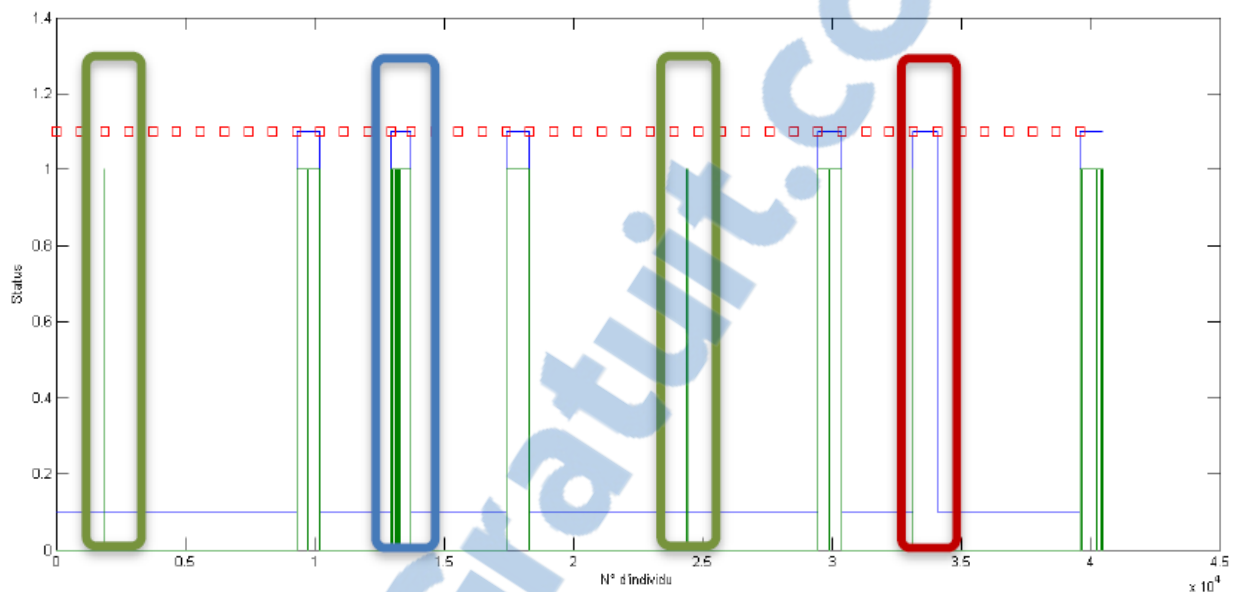


Figure 4.14 Status et Winner pour le générateur 1

Encadré en vert, nous voyons une erreur de prédiction isolée par les forêts aléatoires. Celles-ci peuvent être éliminées avec un retard de décision ou un trigger de Schmitt afin d'éviter que le système soit déclaré instable à chaque mauvaise prédiction.

En rouge par contre, nous voyons que le prédicteur est incapable, sauf pour quatre individus, de détecter la simulation numéro 1 151 comme instable. La figure suivante est un agrandissement de cette simulation.

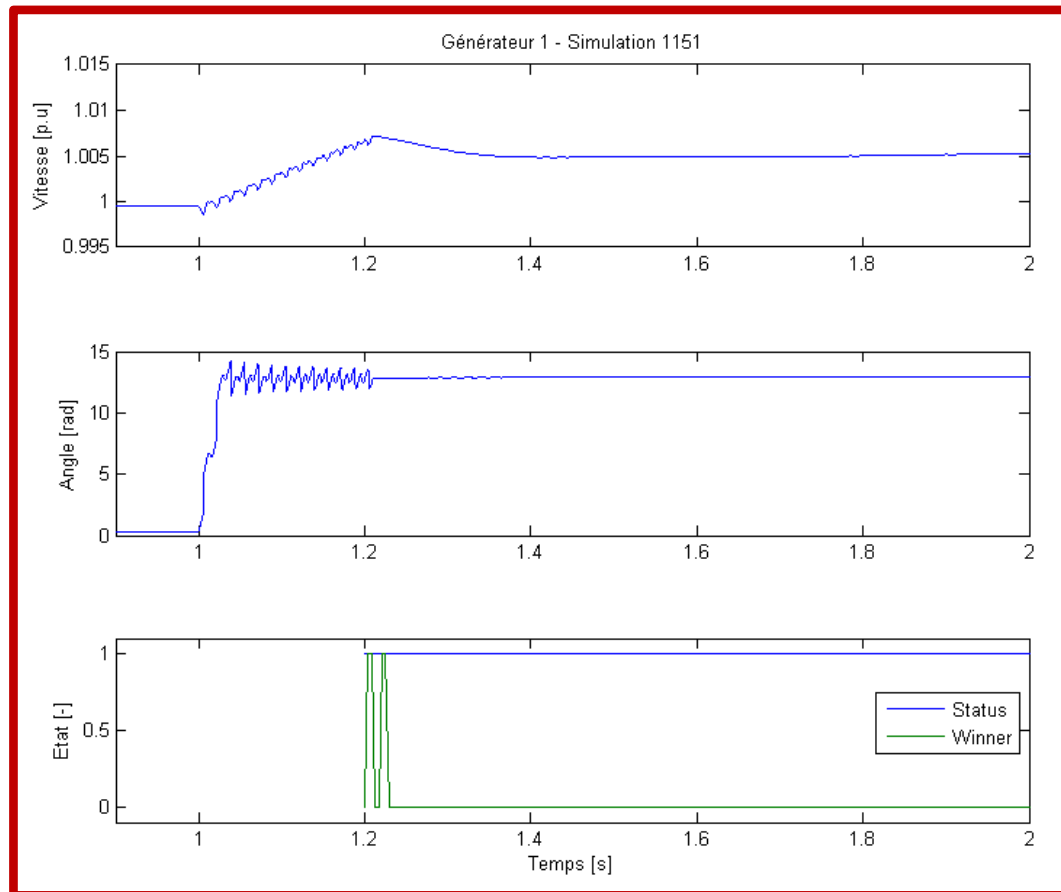


Figure 4.15 Exemple de fonctionnement du RPS pour la simulation 1151 - Générateur 1

Nous voyons que seuls les premiers cas sont déclarés instables et par après, les prédictions sont stables, mais erronées par rapport au COI.

Pour la simulation 1 060, encadrée en bleu, nous voyons qu'au début de l'instabilité, le prédicteur n'est pas sûr et qu'il hésite entre stable et instable au fil des individus, pour au final être sûr de sa décision.

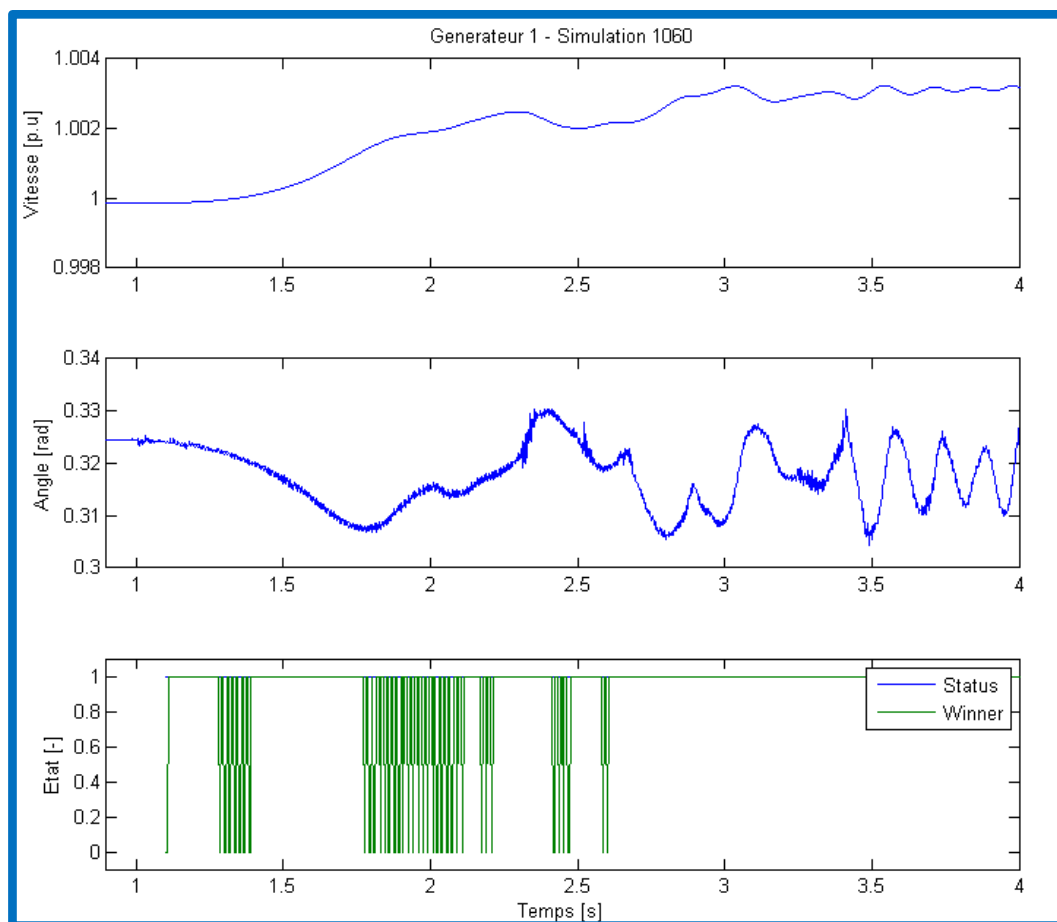


Figure 4.16 Exemple de fonctionnement du RPS pour la simulation 1060 - Générateur 1

Pour les autres cas de simulation, nous voyons que les simulations sont détectées dès les premiers individus et les erreurs pendant l'instabilité sont faibles comme nous le montre la matrice de confusion établie avec l'ensemble de validation.

Actuel /Prédit	0	1
0	35147	18
1	1013	4317

Table 4.18 Matrice de confusion pour le générateur 1 sur l'ensemble de validations



Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de FP	Taux de FN
97.45%	99.95%	80.99%	2.55%	19.01%	0.05%

Table 4.19 Taux mesurés par la matrice de confusions du générateur 1

Nous voyons que le taux d'erreur est faible, mais le taux de faux positifs (1 classé comme 0) est grand. Mettre des pénalités pour les faux positifs serait une solution. Elle augmenterait l'erreur général, mais les cas instables seraient mieux détectés.

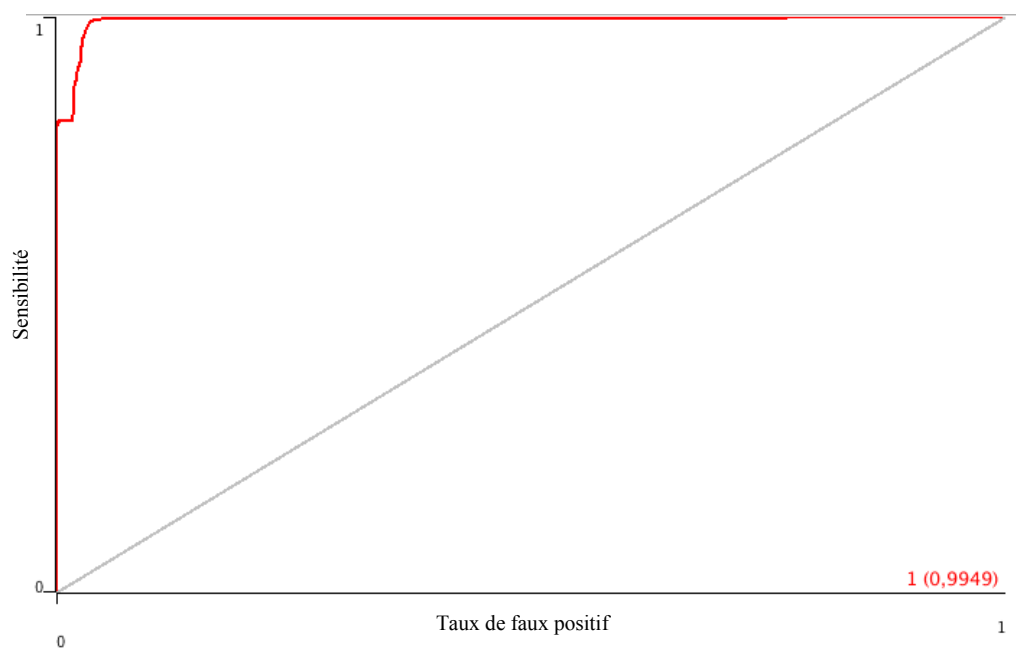


Figure 4.17 Courbe ROC du générateur 1 sur l'ensemble de validation

La pertinence des résultats visuels est confirmée par la courbe ROC ci-dessus. Nous avons une aire sous la courbe de 0.9949, soit une moyenne de 0.9561 sur l'ensemble de validation et de test.

4.4.2 Générateur 2

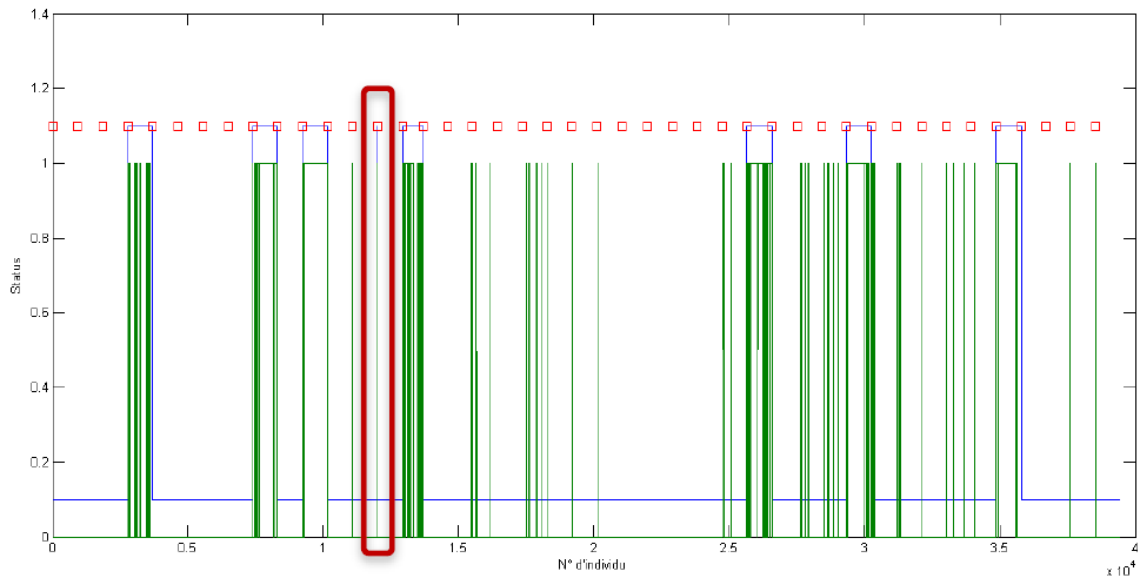


Figure 4.18 Status et Winner pour le générateur 2

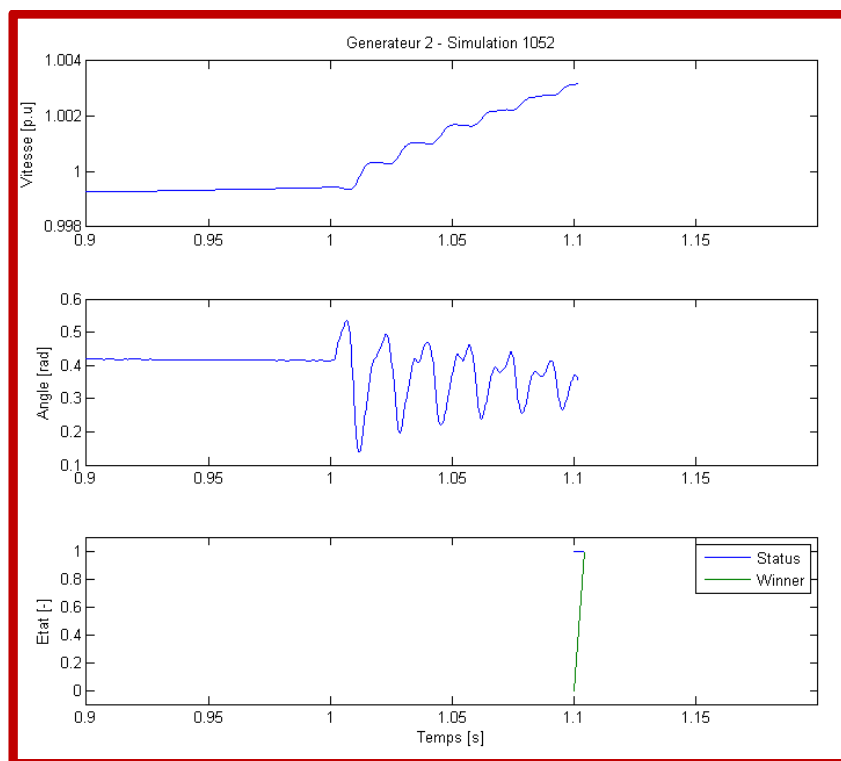


Figure 4.19 Exemple de fonctionnement du RPS pour la simulation 1052 - Générateur 2

En comparaison avec le prédicteur du générateur 1, nous voyons qu'il y a un plus grand nombre d'erreur de prédiction mais qu'il n'y a pas de cas instables non-identifiés. Nous voyons aussi que la simulation 1 052, encadrée en rouge, pour laquelle seulement un individu désigne l'instabilité pour la simulation et il est détecté correctement. Sous EMTP-RV, huit individus ont été sauvegardés pour cette simulation, mais étant donné que sous KNIME, nous avons sous-échantillonné un point sur huit, seul un point reste.

Actuel /Prédit	0	1
0	32941	278
1	1740	4529

Table 4.20 Matrice de confusion pour le générateur 2 sur l'ensemble de validation

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de FP	Taux de FN
94.89%	99.16%	72.24%	5.11%	27.76%	0.84%

Table 4.21 Taux mesurés par la matrice de confusion du générateur 2

Nous voyons que nous avons un taux de faux positifs plus important que pour le générateur 1 ce qui explique que le prédicteur est indécis lors des détections d'instabilités. Le taux de faux négatif n'a augmenté que de 0.8% mais sur la quantité d'individus stables, cette différence d'erreur est non négligeable et explique le nombre d'individus stables classés instables.

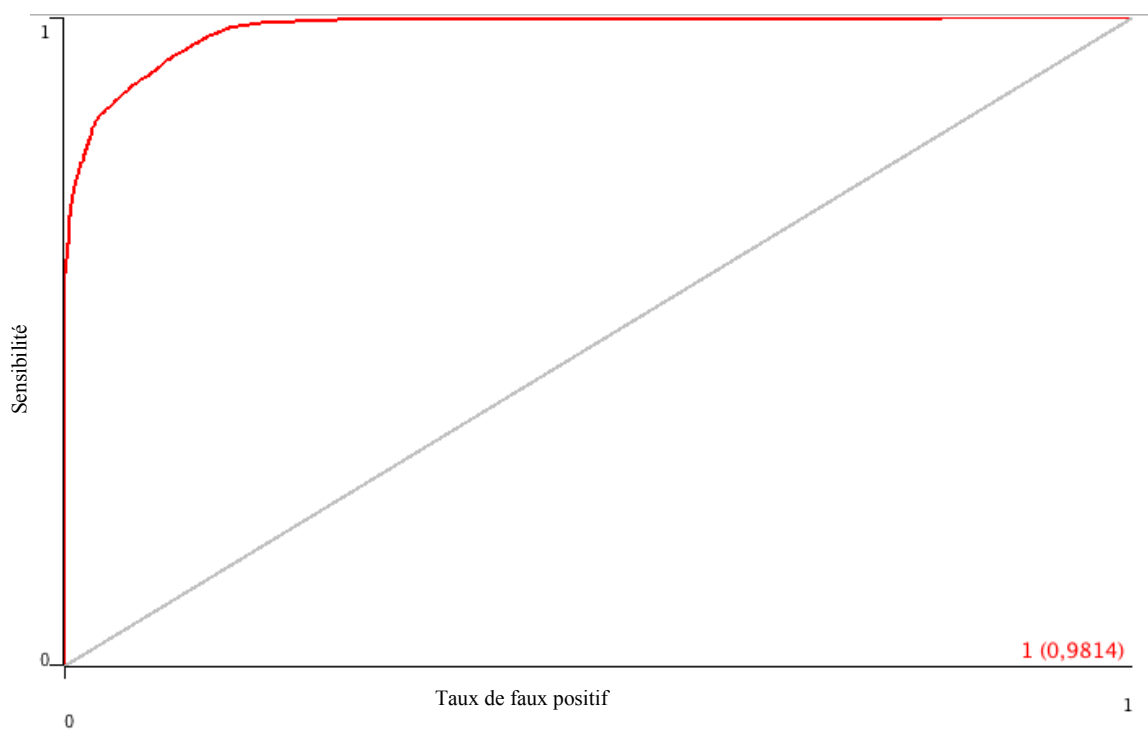


Figure 4.20 Courbe ROC du générateur 2 sur l'ensemble de validation

Nous voyons que nous avons une aire sous la courbe plus faible que pour le générateur 1, ce qui explique les moins bons résultats. En comparaison avec l'ensemble, l'aire sous la courbe de l'ensemble de test, nous voyons que nous avons passé de 0.8659 à 0.9814, soit une moyenne de 0.9237. Cette différence entre ces deux courbes s'explique par la non-homogénéité des deux ensembles.

4.4.3 Générateur 3

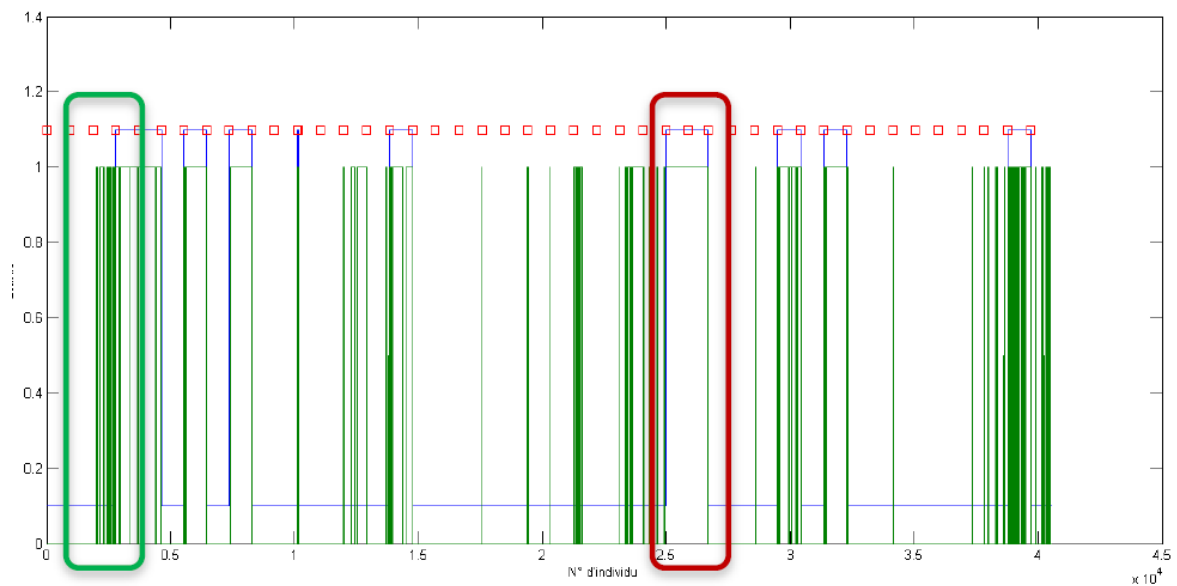


Figure 4.21 Status et Winner pour le générateur 3

Pour le générateur 3, nous voyons que plusieurs simulations, tel par exemple celle encadrée en rouge, sont classées correctement et rapidement et sans indécisions. Par contre, nous voyons aussi que pour plusieurs simulations, telle que celle encadrée en vert, le prédicteur tend à les classer comme instables. Les deux figures suivantes sont un agrandissement de ces deux cas.

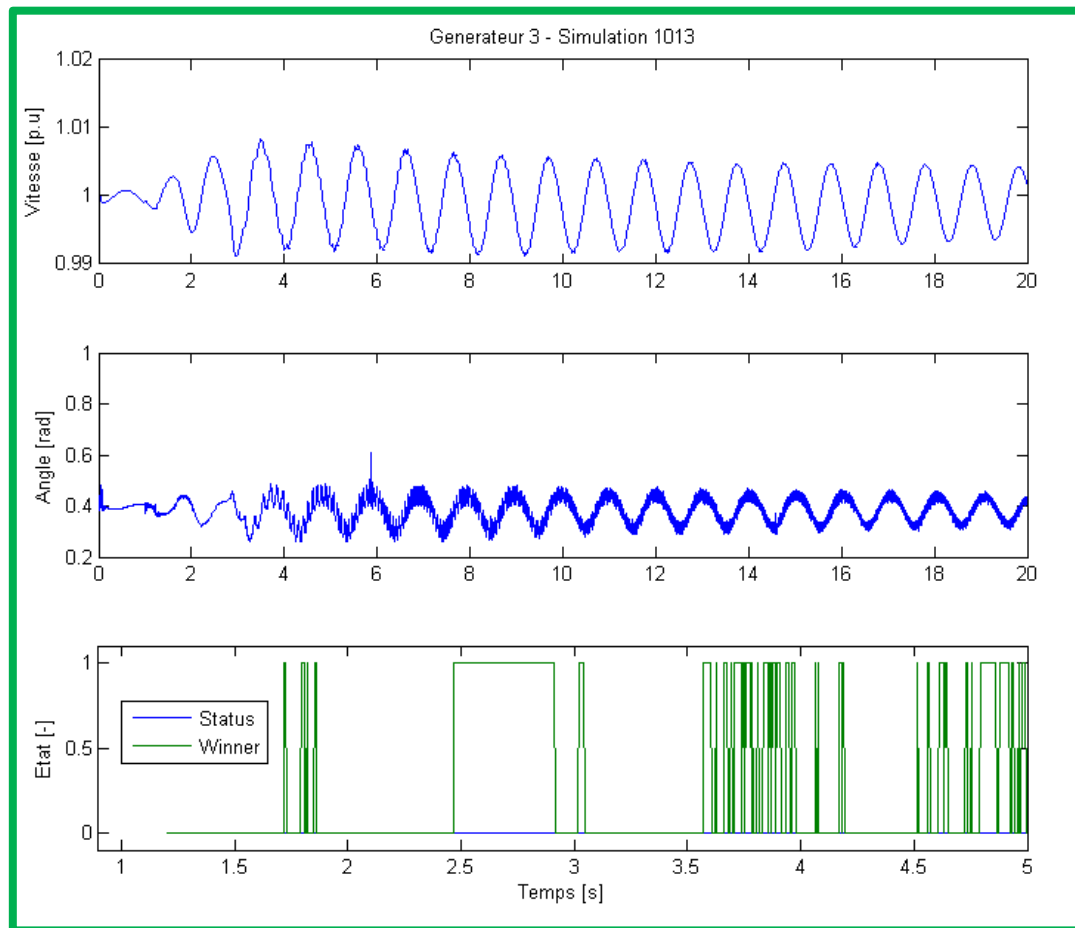


Figure 4.22 Exemple de fonctionnement du RPS pour la simulation 1013 - Générateur 3

Après agrandissement, nous voyons que durant les 3.8 secondes après le court-circuit, le prédicteur a des phases pendant lesquelles il est sur de la stabilité et des phases pendant lesquelles il est sur de l'instabilité.

Dans le cas en rouge par contre, le prédicteur est absolument sur de sa décision et l'instabilité est détectée dès le premier individu échantillonné par les PMU's.

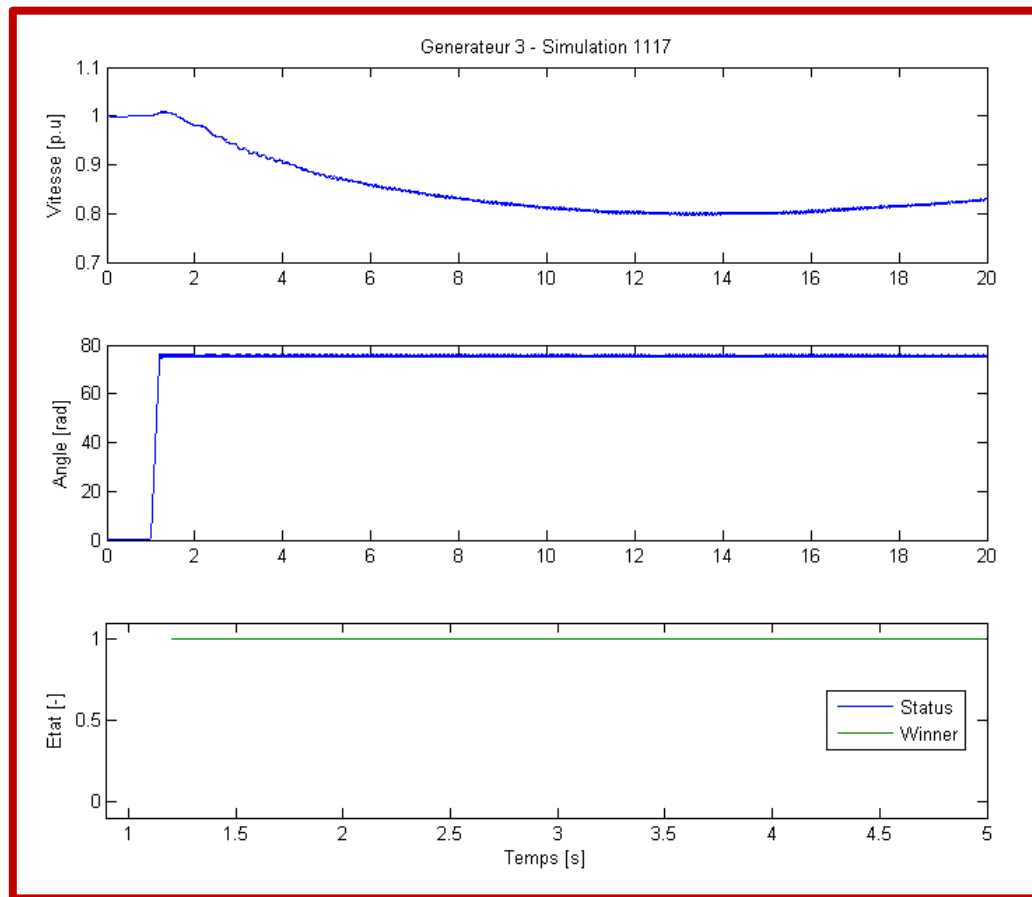


Figure 4.23 Exemple de fonctionnement du RPS pour la simulation 1117 - Générateur 3

En analysant les taux d'erreurs issues de la matrice de confusion, nous voyons que le taux de faux positif est égal au taux de faux négatif. Cependant, étant donné que le nombre de 0 est plus grand, les nombre de mauvaise classification est plus important.

Actuel /Prédit	0	1
0	29614	1830
1	565	8596

Table 4.22 Matrice de confusion pour le générateur 3 sur l'ensemble de validation

Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de FP	Taux de FN
94.10%	94.18%	93.83%	5.90%	6.17%	6.18%

Table 4.23 Taux mesurés par la matrice de confusions du générateur 3

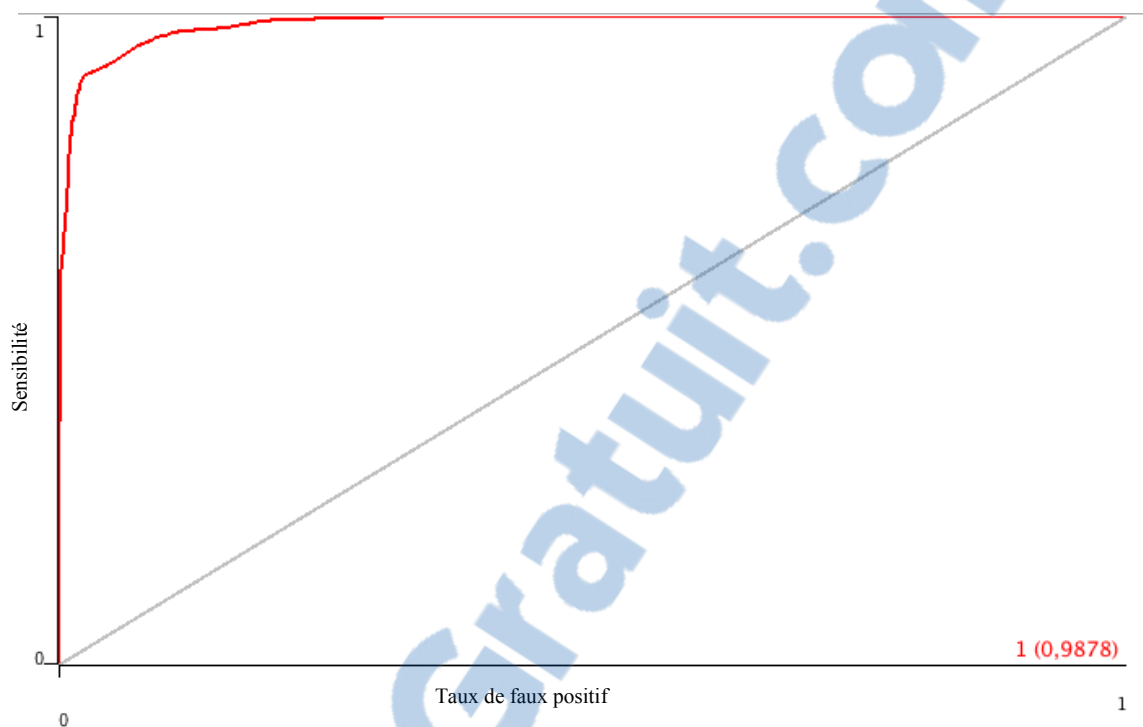


Figure 4.24 Courbe ROC du générateur 3 sur l'ensemble de validation

Pour cet ensemble aussi, nous obtenons de bons résultats, soit une aire sous la courbe de 0.9878 pour l'ensemble de validation. Il est à noter aussi que pour l'ensemble de test, l'aire était de 0.9791, soit une moyenne de 0.9834. Contrairement au générateur 2, l'ensemble de test et de validation du générateur 3 sont homogènes.

4.4.4 Générateur 4

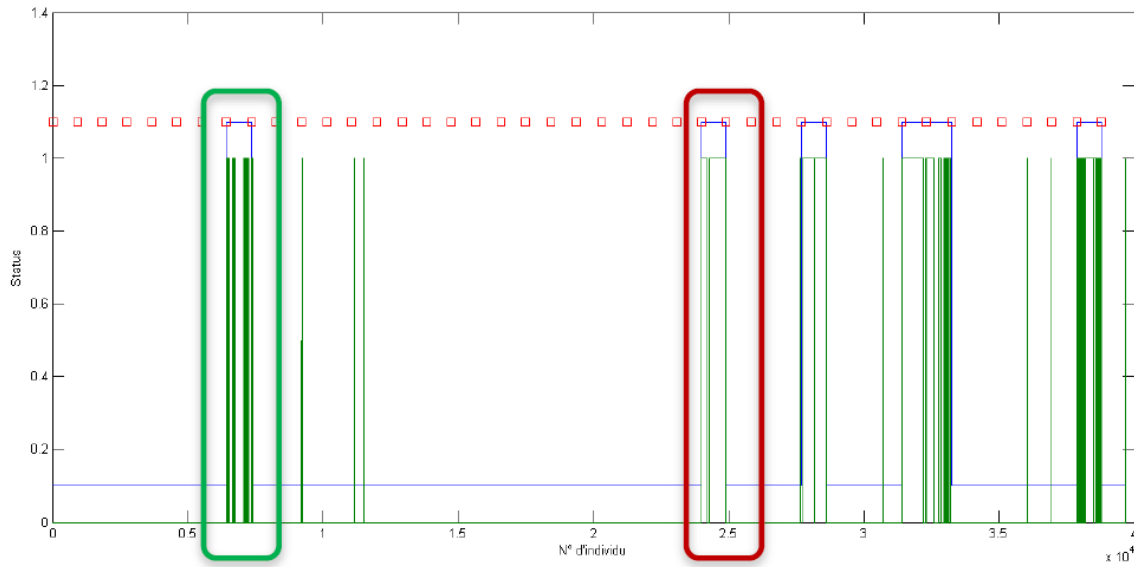


Figure 4.25 Status et Winner pour le générateur 4

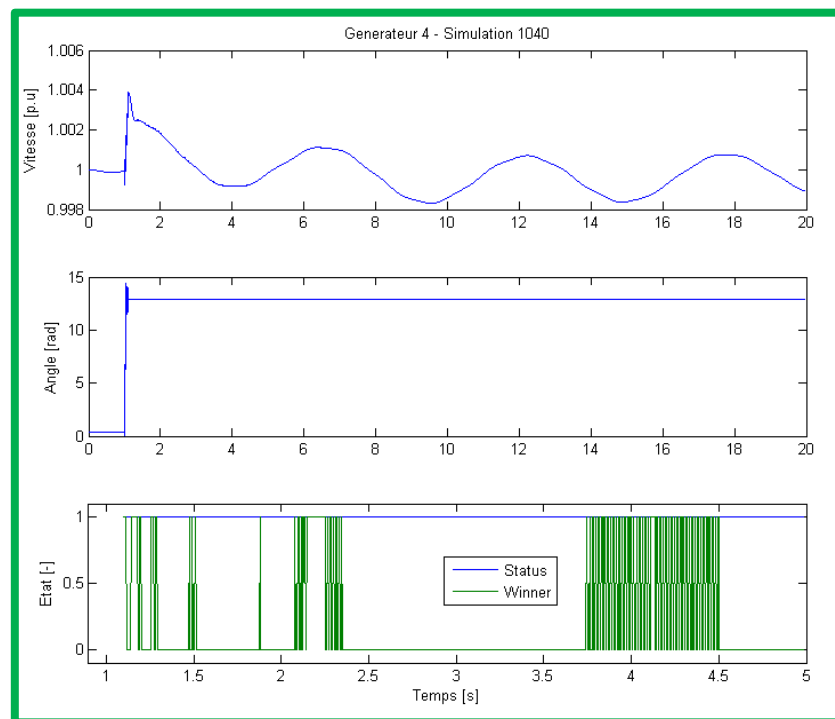


Figure 4.26 Exemple de fonctionnement du RPS pour la simulation 1040 - Générateur 4

En analysant la figure 4.26, nous voyons que le prédicteur a tendance à classifier le cas comme stable alors que celui-ci est instable selon le RPS. Pendant certaines périodes, celui-ci n'est pas convaincu et oscille entre stable et instable tel que pour $t > 3.7$ et $t < 4.5$.

Pour le cas en rouge de la figure ci-dessous, le prédicteur est sûr de la stabilité et de l'instabilité.

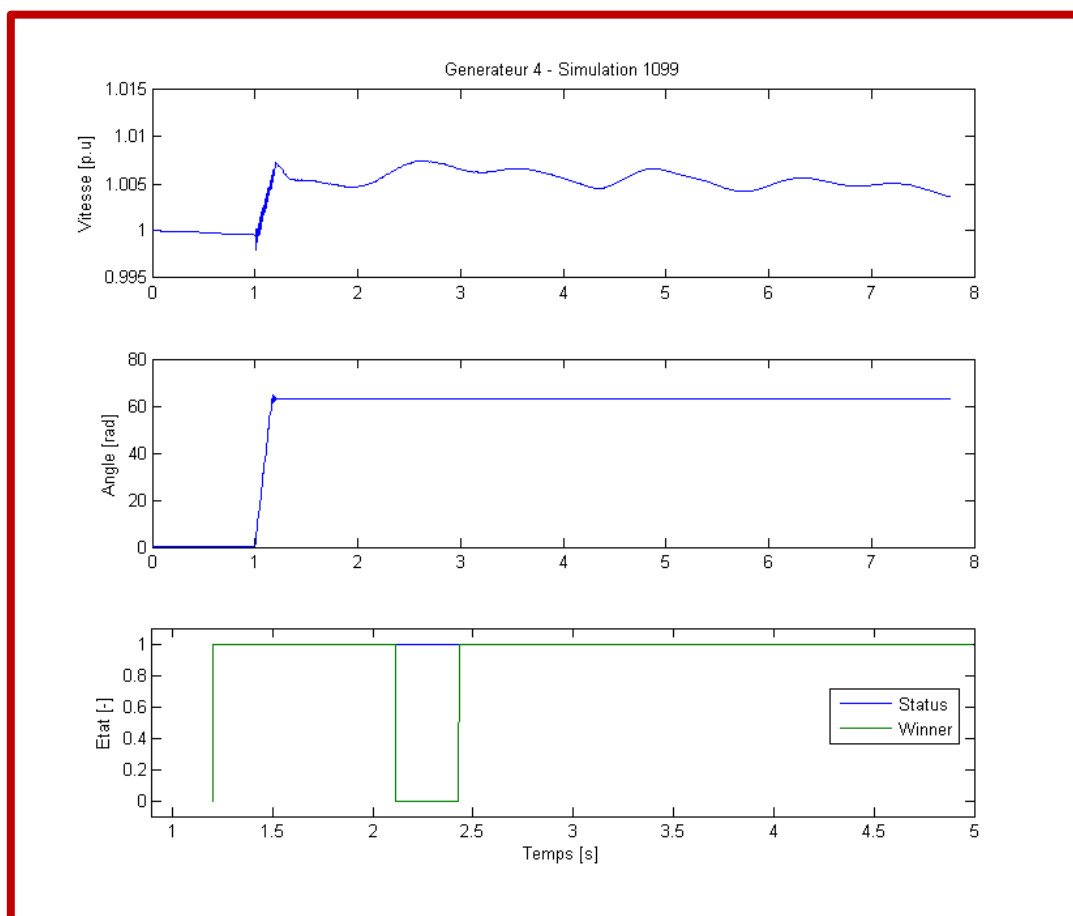


Figure 4.27 Exemple de fonctionnement du RPS pour la simulation 1099 - Générateur 4

Pour le générateur 4, nous voyons qu'il y'a quelques cas stables qui sont déclarés instables. Par contre, concernant les cas instables, nous voyons des simulations, en rouge, qui sont détectées nettement, et d'autres simulations, en vert, où le prédicteur est plus indécis.



Atuel /Prédit	0	1
0	34100	56
1	1568	3978

Table 4.24 Matrice de confusion pour le générateur 4 sur l'ensemble de validation

Précision	Sensibilité	Specificité	Taux d'erreur	Taux de FP	Taux de FN
95.91%	99.84%	71.73%	4.09%	28.27%	0.16%

Table 4.25 Taux mesurés par la matrice de confusions du générateur 4

Finalement, à la vue de ces quatre figures, nous pouvons voir que les instabilités sont mieux détectées pour les gros générateurs que les petits.

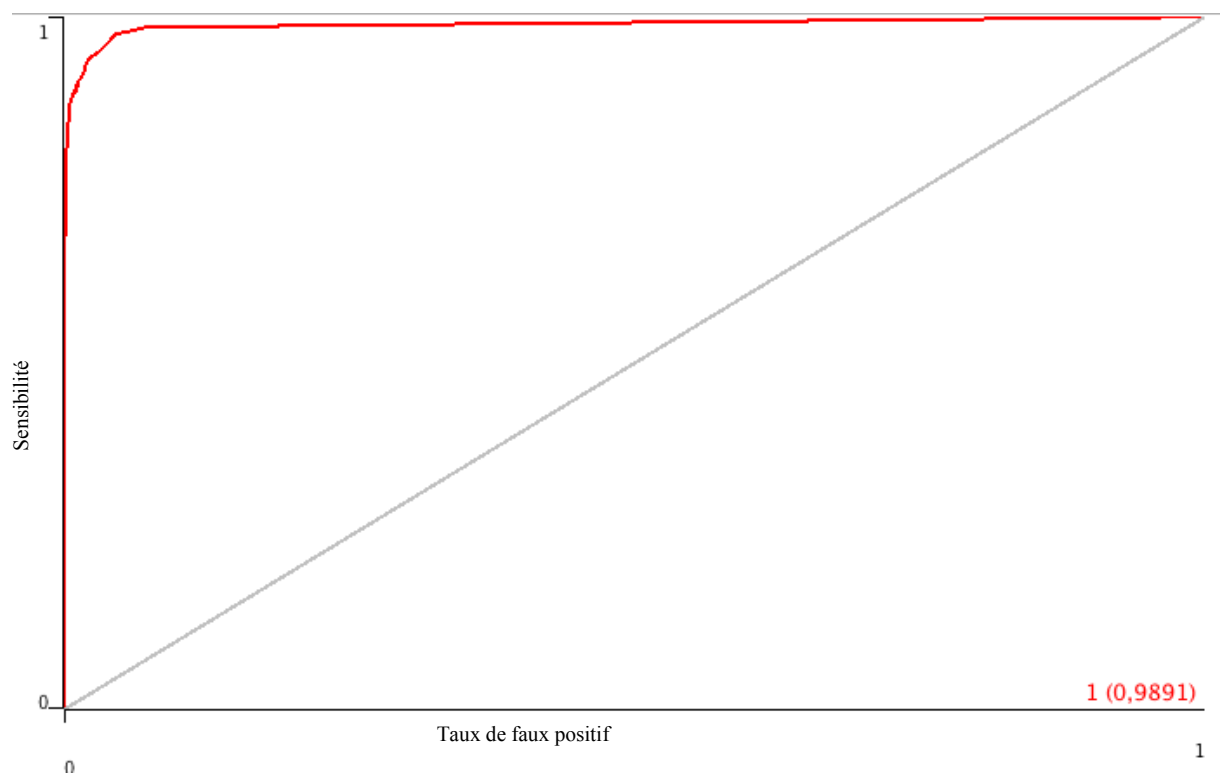


Figure 4.28 Courbe ROC du générateur 4 sur l'ensemble de validation

Finalement, pour le générateur 4 aussi la qualité de prédiction est bonne avec une aire sous la courbe de 0.9891 soit en moyenne, 0.9732.

Pour les quatre générateurs, nous obtenons une aire sous la courbe moyenne de 0.9591 sur l'ensemble de test et de validation. Ce résultat est représentatif de la capacité de généralisation du prédicteur car ces deux sous-ensembles représentent 40% des simulations.

4.5 Réduction du taux de FP

Ce sous-chapitre présente la réduction du taux de faux positif du prédicteur.

Selon [25], le taux d'erreur d'un prédicteur dépend de la formule suivante, soit l'erreur sur les cas stables additionnée à l'erreur sur les cas instables.

$$\varepsilon(DT, F) = \sum_{(x,y) \in F_{Stable}} (1 - ip)\lambda + \sum_{(x,y) \in F_{Instables}} ip\lambda \quad (4.1)$$

Où ε est l'erreur de l'arbre de décision DT , F est le jeu de donnée entier, λ est le risque d'erreur de classification, p est la part de cas instables.

Lorsque la variable à prédire n'est pas équilibrée il est possible de donner plus d'importance à une classe en faveur de l'autre en la multipliant i -fois et en l'ajoutant dans l'ensemble d'apprentissage. L'idéal est une répartition de : 50% d'individu stables et 50% d'individus instables.

Dans notre cas, étant donné que nous avons plus de cas stables que de cas instables, nous quadruplons les cas instables afin d'avoir un nombre d'individus de chaque classe égal.

Les tableaux suivants présentent les résultats obtenus par le prédicteur lorsque la base de données d'apprentissage est standard ou équilibrée. Les résultats sont obtenus en testant le prédicteur sur l'ensemble de test.

100 RF

		Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de FP	Taux de FN	% d'amélioration de FP
Gen 1	Equilibrée	97.45	99.95	80.99	2.55	19.01	0.05	-5.48 %
	Standard	97.34	99.99	79.89	2.66	20.11	0.01	
Gen 2	Equilibrée	94.85	99.17	71.96	5.15	28.04	0.84	-1.18 %
	Standard	94.91	99.30	71.62	5.09	28.38	0.70	
Gen 3	Equilibrée	94.98	95.40	93.52	5.02	6.48	4.82	0.68 %
	Standard	94.59	94.90	93.56	5.41	6.44	5.38	
Gen 4	Equilibrée	95.91	99.84	71.73	4.09	28.27	0.16	-1.70 %
	Standard	95.88	99.88	71.24	4.12	28.76	0.12	

Table 4.26 Taux de 100 RF pour une base de données Standard et équilibrée

10 RF

		Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de FP	Taux de FN	% d'amélioration de FP
Gen 1	Equilibrée	97.02	99.99	77.43	2.98	22.57	0.01	6.55 %
	Standard	97.2	99.99	78.82	2.8	21.18	0.01	
Gen 2	Equilibrée	94.47	99.54	67.63	5.53	32.37	0.46	9.03 %
	Standard	94.55	99.12	70.31	5.45	29.69	0.89	
Gen 3	Equilibrée	94.21	95.56	89.56	5.79	10.44	4.65	16.02 %
	Standard	94.78	95.88	91.01	5.22	8.99	4.3	
Gen 4	Equilibrée	96.08	99.76	73.4	3.92	26.6	0.24	-10.66 %
	Standard	95.76	99.9	70.23	4.24	29.77	0.1	

Table 4.27 Taux de 10 RF pour une base de données Standard et équilibrée

2 RF

		Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de FP	Taux de FN	% d'amélioration de FP
Gen 1	Équilibrée	95.26	99.26	68.82	4.74	31.18	0.74	3.94 %
	Standard	96.04	99.98	70	3.96	30	0.02	
Gen 2	Équilibrée	90.53	99.23	44.42	9.47	55.58	0.78	43.79 %
	Standard	93.21	99.22	61.35	6.79	38.65	0.78	
Gen 3	Équilibrée	91.35	96.86	72.44	8.65	27.56	3.24	40.9 %
	Standard	92.99	96.64	80.44	7.01	19.56	3.47	
Gen 4	Équilibrée	95.26	99.77	67.54	4.74	32.46	0.23	-15.53 %
	Standard	94.61	99.98	61.58	5.39	38.42	0.02	

Table 4.28 Taux de 2 RF pour une base de données Standard et équilibrée

En comparant les 3 tableaux précédents, nous voyons que plus la forêt aléatoire est grande plus les résultats sont stables et ne sont pas affectés par le déséquilibre des classes de la variable à prédire.

Nous voyons aussi que si nous équilibrons la base de données il peut y avoir un effet contraire à l'effet désiré comme sur le tableau 4.27 où le taux de faux positif est accru de 40.9 et 43.79 % par rapport à la base de données originale. Par conséquent, cette méthode de réduction du taux de faux positifs est à éviter pour les forêts aléatoires.

ADABOOST

		Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de FP	Taux de FN	% d'amélioration de FP
Gen 1	Équilibrée	94.07	94.62	90.43	5.93	9.57	5.68	-82.94 %
	Standard	92.62	100	43.92	7.38	56.08	0	
Gen 2	Équilibrée	87.75	92.64	61.83	12.25	38.17	7.95	-31.16 %
	Standard	89	97.39	44.55	11	55.45	2.68	
Gen 3	Équilibrée	78.51	82.23	65.76	21.49	34.24	21.61	-27.13 %
	Standard	86.78	96.61	53.01	13.22	46.99	3.51	
Gen 4	Équilibrée	86.24	85.13	93.08	13.76	6.92	17.47	-81.74 %
	Standard	89.23	93.64	62.08	10.77	37.92	6.79	

Table 4.29 Taux pour ADABOOST pour une base de données Standard et équilibrée

ADTree

		Précision	Sensibilité	Spécificité	Taux d'erreur	Taux de FP	Taux de FN	% d'amélioration de FP
Gen 1	Equilibrée	99.7	99.84	98.74	0.3	1.26	0.16	-94.64 %
	Standard	96.89	99.97	76.57	3.11	23.43	0.03	
Gen 2	Equilibrée	92.5	92.75	91.23	7.5	8.77	7.82	-74.65 %
	Standard	90.37	95.08	65.39	9.63	34.61	5.17	
Gen 3	Equilibrée	85.63	87.62	78.82	14.37	21.18	14.13	-44.44 %
	Standard	88.35	96.06	61.88	11.65	38.12	4.1	
Gen 4	Equilibrée	86.24	85.13	93.08	13.76	6.92	17.47	-73.59 %
	Standard	90.24	92.91	73.78	9.76	26.22	7.63	

Table 4.30 Taux pour ADTree pour une base de données Standard et équilibrée

Concernant les prédicteurs de type ADABOOST et ADTree, nous voyons grâce aux deux tableaux précédents que cette méthode est efficace, car elle permet effectivement de réduire le taux de faux positifs, avec comme inconvénient d'augmenter le taux de faux négatif.

Conclusion

L'objectif de cette recherche était de savoir s'il était possible et faisable de concevoir un relais de protection de synchronisme des réseaux électriques en temps réel en utilisant l'exploration de données et la famille des arbres de décision.

Pour en tester sa faisabilité, 180 simulations ont été réalisées en utilisant EMTP-RV, en utilisant le réseau Anderson comme système d'étude et EMTP-RV comme logiciel. 180 simulations de vingt secondes et avec une fréquence d'échantillonnage de 3840 [Hz] sont effectuées en variant le load-flow, le point d'impact du court-circuit, le type et la durée de celui-ci. Pour chacune des simulations, 41 mesures sont prises par générateur, soit 39 mesures électriques tels que les tensions de chaque phase temporelle et RMS, les courants de phase temporelle et RMS, les fréquences, les diverses puissances actives et réactives ainsi que deux mesures mécaniques que sont l'angle de charge et la vitesse de rotation.

Les simulations sont exportées par des fichiers de type .MAT sous MATLAB dans lequel les données sont tout d'abord converties dans un format simple et les individus électriques supérieurs à cinq secondes sont supprimés à cause de la limitation en espace disque et de mémoire RAM.

Les simulations ayant été arrêtées par EMTP-RV avant la fin du court-circuit ne sont pas prises en compte car le relais de protection de synchronisme commence à classifier les individus stables et instables après la fin de la contingence.

33 nouvelles variables telles que les fréquences inter-génératrices, la dérivé de la fréquence, le SVC et dSVC/dt sont calculées en utilisant les 39 variables que nous fournit EMTP-RV.

En utilisant le centre d'inertie d'angle et de vitesse, les simulations ont été classifiées comme stables et instables pour chaque générateur. Le tableau ci-dessous en résume la quantité :

	Générateur 1	Générateur 2	Générateur 3	Générateur 4
Stables	152	143	136	151
Instables	26	35	42	27

Les petits générateurs 2 et 3 ont tendance à diverger plus souvent car leur constante d'inertie est plus faible et ils ne sont pas en mesure de maintenir le synchronisme.

La méthode de classification est rapide et intuitive, mais pas optimale. Les figures montrent qu'il y a des simulations pour lesquelles leur vitesse croît de manière exponentielle sans que le centre d'inertie d'angle et de vitesse ne les classifie comme instables. Les données sont exportées depuis MATLAB vers KNIME en utilisant un format .CSV avec entêtes.

Pour le générateur 1, une comparaison des modèles prédictifs, tels que C4.5, CART, ADABOOST, ADTree, et forêts aléatoires est effectuée en utilisant l'aire sous la courbe comme critère de comparaison. De plus, l'aire sous la courbe de ces modèles prédictifs a été calculée pour une fréquence d'échantillonnage de 1920, 960, 480, 240, 120, 60 et 30 [Hz]. Pour les arbres de décision classiques tels que CART ou C4.5, leur aire sous la courbe varie fortement car la qualité de prédiction de ces modèles est fortement reliée à la variance de la base de données. Ces mêmes résultats montrent que les arbres de décision ADTree et forêts aléatoires offrent l'aire sous la courbe la plus constante et la plus élevée grâce aux méthodes de boosting et bagging. Finalement, une fréquence d'échantillonnage de 240 [Hz] offre un bon compromis entre aire sous la courbe et ressources informatiques.

Les 288 variables de tous les générateurs sont filtrées en utilisant les méthodes :

- Symmetrical Uncert Attribute Set Eval avec l'algorithme de recherche Fast Correlation Based Filter. Cette méthode ne garde que dix variables, mais c'est celle qui offre la plus mauvaise qualité de prédiction ;
- Consistency Subset Eval avec l'algorithme de recherche Best First et Rank Search garde 28 variables et respectivement 72 variables et offre des performances quasi égales au 288 variables ;
- Correlation-based Feature Subset Selection avec l'algorithme de recherche Best First et Rank Search garde 60 variables et 94 variables et offre des performances légèrement plus faibles que Consistency Subset Eval.

Au final, le modèle de prédiction forêt aléatoire est choisi et la méthode de filtrage Consistency Subset Eval avec l'algorithme de recherche Best First est choisi car c'est celui qui offre la meilleure qualité de prédiction moyenne et un bon compromis entre nombre de variables et qualité de prédiction.

Après avoir filtré les variables et échantillonné à 240 [Hz], l'ensemble de validation est soumis à la forêt aléatoire. Les résultats montrent que pour les quatre machines, nous obtenons des aires sous la courbe de 0.9949, 0.9814, 0.9878 et 0.9871, soit une aire moyenne de 0.9883 pour les quatre générateurs.

En prenant la moyenne des aires sous la courbe de l'ensemble de test et de validation qui représentent 60% des simulations, nous obtenons une aire sous la courbe de 0.9591. Cela est suffisant pour conclure que les forêts aléatoires ont appris et généralisé correctement l'ensemble d'apprentissage à l'ensemble de test et de validation et par conséquent, la réalisation d'un relais de protection de synchronisme en temps réel est possible.

Une tentative de réduction du taux de faux positif a été effectuée en équilibrant les classes de l'ensemble d'apprentissage. Les résultats ont montré que cette méthode est inefficace

pour les forêts aléatoires, mais qu'elle permet une réduction significative du nombre de faux positifs au profit des faux négatifs pour les arbres de type ADABOOST et ADTree.

Perspectives de recherches

Un point très important à améliorer est la discrimination des cas stables et instables en utilisant par exemple le Contingency Clearing Time, la matrice d'état de tout le système ou encore en utilisant comme critère le nombre de tours d'angles internes ou les inversions de puissance, comme le distributeur français RTE [21].

Étant donné que nous avons déterminé une fréquence d'échantillonnage plus raisonnable, il serait intéressant d'augmenter la quantité de simulations en faisant d'autres load-flows par exemple ou en modifiant le type de court-circuit ou encore en faisant un changement soudain de charge ou de topologie du réseau. Il serait aussi intéressant de voir l'efficacité et l'apport de nouvelles variables telles que les ondelettes.

Il serait aussi intéressant de créer des simulations qui sont très violentes et qui résultent en une instabilité avant la fin du court-circuit et par conséquent, le modèle prédictif pourrait fonctionner tout le temps.

Une autre perspective de recherche pourrait être de créer un modèle de décision par type de court-circuit ou par type de load-flow et ainsi savoir s'il y a amélioration des résultats.

Dans ce document, nous avons choisi de classer individu par individu l'état de ceux-ci. Par conséquent, nous ne pouvons pas normaliser, réduire leur variance et utiliser des méthodes de composantes principales etc. Pour remédier à cela, il pourrait être judicieux de tester par exemple la stabilité d'un système par pas de 0.1 ou 0.5 secondes. Entre temps, les données seraient accumulées et la prise de décision ultime se ferait après qu'une quantité suffisante d'individus y soit entrée.

Bibliographie

- [1] S. Russell et P. Norvig, Artificial Intelligence: A Modern Approach, 3rd edition, Prentice Hall, 2009.
- [2] L. Breiman, J. H. Friedman, R. A. Olshen et C. J. Stone, Classification and regression trees, Monterey: Wadsworth & Brooks/Cole Advanced Books & Software, 1984.
- [3] L. Breiman, «Random Forest, Machine Learning, vol. 45, pp. 5-32,» 2001.
- [4] P. Kundur, Power Systems Stability and Control, McGraw-Hill Professional , 1994.
- [5] P. S. R. C. o. t. I. P. E. IEEE, «POWER SWING AND OUT-OF-STEP CONSIDERATIONS ON TRANSMISSION LINES,» 2005.
- [6] L. Wehenkel, T. Van Cutsem et M. Ribbens-Pavella, Decision Trees applied to on-line Transient Stability Assessment of Power Systems, Liège: Université de Liège, 1988.
- [7] H. Hashim, I. Abidin, K. S. Yap, I. Musirin et M. Zulkepali, «An Analysis of Transient Stability Using Center-of-Inertia: Angle and Speed,» Power and Energy

- (PECon), 2010 IEEE International Conference on , 2010.
- [8] L. Rokach et O. Maimon, Data Mining with Decision Trees: Theory and Applications, World Scientific Publishing Co Pte Ltd, 2008.
 - [9] S. Tufféry, Data mining et statistique décisionnelle, Editions TECHNIP, 2005.
 - [10] Y. Freund et R. Schapire, «A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,» 1996.
 - [11] I. Witten et F. Eibe, Data Mining Practical Machine Learning Tools and Techniques 2nd Edition, Morgan Kaufmann, 2005.
 - [12] T. Kalafatis, «So...What's important??,» Octobre 2008. [En ligne]. Available: <http://lifeanalytics.blogspot.com/2008/10/sowhats-important.html>. [Accès le Février 2011].
 - [13] I. Kojadinovic et T. Wotzka, «Comparison between a filter and a wrapper approach to variable subset selection in regression problems,» ESIT 2000, Aachen, 2000.
 - [14] M. Hall, «Correlation-based Feature Selection for,» University of Waikato, Waikato, Avril 1999.
 - [15] L. Yu et H. Liu, «Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution,» Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington D.C., 2003.

- [16] S. Rudy et H. Liu, A probabilistic approach to feature selection- A filter solution, Singapore: University of Singapore, 1996.
- [17] M. Gütlein, F. Eibe, M. Hall et A. Karwath, «Large-scale attribute selection using wrappers,» Freiburg, Germany, 2006.
- [18] P. M. Anderson et R. G. Farmer, Series Compensation of Power Systems, Encinitas, California: PBLSH, 1996.
- [19] H. Bakashi, «Control and protection of distribution network with non-utility induction generators using EMTP-RV,» Université Laval, 2006.
- [20] P. W. Sauer et M. A. Pai, «Power System Dynamics and Stability,» Prentice Hall, 1998.
- [21] «Référentiel Technique,» RTE, mai 2008. [En ligne]. Available: http://clients.rte-france.com/htm/fr/mediatheque/telecharge/reftech/15-05-08_article_4-3__v3.pdf. [Accès le Février 2012].
- [22] I. Kamwa, R. Grondin, M. Dubrescu et M. Rousseau, «Relais de Perte de Synchronisme - RPS Analyse des performances de l'algorithme,» Québec, Canada, 2003.
- [23] K. Brady, «CSV with column headers,» MATLAB CENTRAL, 6 Janvier 2011. [En ligne]. Available: <http://www.mathworks.com/matlabcentral/fileexchange/29933-csv-with-column-headers>. [Accès le Janvier 2012].

- [24] R. Rakotomalala, «Comparaison des performances des logiciel libres,» Université de Lyon, 20 Octobre 2011. [En ligne]. Available: http://eric.univ-lyon2.fr/~ricco/tanagra/fichiers/fr_Tanagra_Perfs_Comp_Decision_Tree_Suite.pdf. [Accès le Janvier 2012].
- [25] I. Kamwa, S. Samantaray et G. Joos, Catastrophe Predictors From Ensemble Decision-Tree Learning of Wide-Area Severity Indices, IEEE Transactions on Smart Grid, Sept 2010.
- [26] D.-T. Larose et T. Vallaud, Des données à la connaissance : Une introduction au data-mining, Vuibert, 2005.

Annexe 1

Noms des simulations lorsque le générateur 1 est défini comme Slack.

'rpsslackg1_1001_tripahse_ligne30_long.mat'	'rpsslackg1_1031_biphase_barreb1_long.mat'
'rpsslackg1_1002_tripahse_ligne30_court.mat'	'rpsslackg1_1032_biphase_barreb1_court.mat'
'rpsslackg1_1003_tripahse_ligne30_long.mat'	'rpsslackg1_1033_biphase_barreb2_long.mat'
'rpsslackg1_1004_tripahse_ligne30_court.mat'	'rpsslackg1_1034_biphase_barreb2_court.mat'
'rpsslackg1_1005_tripahse_ligne30_long.mat'	'rpsslackg1_1035_biphase_barreb3_long.mat'
'rpsslackg1_1006_tripahse_ligne30_court.mat'	'rpsslackg1_1036_biphase_barreb3_court.mat'
'rpsslackg1_1007_tripahse_ligne30_long.mat'	'rpsslackg1_1037_biphase_barreb4_long.mat'
'rpsslackg1_1008_tripahse_ligne30_court.mat'	'rpsslackg1_1038_biphase_barreb4_court.mat'
'rpsslackg1_1009_tripahse_ligne30_court.mat'	'rpsslackg1_1039_biphase_barreb5_long.mat'
'rpsslackg1_1010_tripahse_ligne30_court.mat'	'rpsslackg1_1040_biphase_barreb5_court.mat'
'rpsslackg1_1011_tripahse_barreb1_long.mat'	'rpsslackg1_1041_monophasse_ligne30_long.mat'
'rpsslackg1_1012_tripahse_barreb1_court.mat'	'rpsslackg1_1042_monophasse_ligne30_court.mat'
'rpsslackg1_1013_tripahse_barreb2_long.mat'	'rpsslackg1_1043_monophasse_ligne30_long.mat'
'rpsslackg1_1014_tripahse_barreb2_court.mat'	'rpsslackg1_1044_monophasse_ligne30_court.mat'
'rpsslackg1_1015_tripahse_barreb3_long.mat'	'rpsslackg1_1045_monophasse_ligne30_long.mat'
'rpsslackg1_1016_tripahse_barreb3_court.mat'	'rpsslackg1_1046_monophasse_ligne30_court.mat'
'rpsslackg1_1017_tripahse_barreb4_long.mat'	'rpsslackg1_1047_monophasse_ligne30_long.mat'
'rpsslackg1_1018_tripahse_barreb4_court.mat'	'rpsslackg1_1048_monophasse_ligne30_court.mat'
'rpsslackg1_1019_tripahse_barreb5_long.mat'	'rpsslackg1_1049_monophasse_ligne30_long.mat'
'rpsslackg1_1020_tripahse_barreb5_court.mat'	'rpsslackg1_1050_monophasse_ligne30_court.mat'
'rpsslackg1_1021_biphase_ligne30_long.mat'	'rpsslackg1_1051_monophasse_barreb1_long.mat'
'rpsslackg1_1022_biphase_ligne30_court.mat'	'rpsslackg1_1052_monophasse_barreb1_court.mat'
'rpsslackg1_1023_biphase_ligne30_long.mat'	'rpsslackg1_1053_monophasse_barreb2_long.mat'
'rpsslackg1_1024_biphase_ligne30_court.mat'	'rpsslackg1_1054_monophasse_barreb2_court.mat'
'rpsslackg1_1025_biphase_ligne30_long.mat'	'rpsslackg1_1055_monophasse_barreb3_long.mat'
'rpsslackg1_1026_biphase_ligne30_court.mat'	'rpsslackg1_1056_monophasse_barreb3_court.mat'
'rpsslackg1_1027_biphase_ligne30_long.mat'	'rpsslackg1_1057_monophasse_barreb4_court.mat'
'rpsslackg1_1028_biphase_ligne30_court.mat'	'rpsslackg1_1058_monophasse_barreb4_long.mat'
'rpsslackg1_1029_biphase_ligne30_long.mat'	'rpsslackg1_1059_monophasse_barreb5_long.mat'
'rpsslackg1_1030_biphase_ligne30_court.mat'	'rpsslackg1_1060_monophasse_barreb5_court.mat'

Noms des simulations lorsque le générateur 3 est défini comme Slack.

'rpsslackg3_1061_tripahse_lignea_long.mat'	'rpsslackg3_1091_biphase_barreb1_long.mat'
'rpsslackg3_1062_tripahse_lignea_court.mat'	'rpsslackg3_1092_biphase_barreb1_court.mat'
'rpsslackg3_1063_tripahse_ligne_b_long.mat'	'rpsslackg3_1093_biphase_barreb2_long.mat'
'rpsslackg3_1064_tripahse_ligne_b_court.mat'	'rpsslackg3_1094_biphase_barreb2_court.mat'
'rpsslackg3_1065_tripahse_lignec_long.mat'	'rpsslackg3_1095_biphase_barreb3_long.mat'
'rpsslackg3_1066_tripahse_lignec_court.mat'	'rpsslackg3_1096_biphase_barreb3_court.mat'
'rpsslackg3_1067_tripahse_ligned_long.mat'	'rpsslackg3_1097_biphase_barreb4_long.mat'
'rpsslackg3_1068_tripahse_ligned_court.mat'	'rpsslackg3_1098_biphase_barreb4_court.mat'
'rpsslackg3_1069_tripahse_ligne_f_long.mat'	'rpsslackg3_1099_biphase_barreb5_long.mat'
'rpsslackg3_1070_tripahse_ligne_f_court.mat'	'rpsslackg3_1100_biphase_barreb5_court.mat'
'rpsslackg3_1071_tripahse_barreb1_long.mat'	'rpsslackg3_1101_monophase_lignea_long.mat'
'rpsslackg3_1072_tripahse_barreb1_court.mat'	'rpsslackg3_1102_monophase_lignea_court.mat'
'rpsslackg3_1073_tripahse_barreb2_long.mat'	'rpsslackg3_1103_monophase_ligne_b_long.mat'
'rpsslackg3_1074_tripahse_barreb2_court.mat'	'rpsslackg3_1104_monophase_ligne_b_court.mat'
'rpsslackg3_1075_tripahse_barreb3_long.mat'	'rpsslackg3_1105_monophase_lignec_long.mat'
'rpsslackg3_1076_tripahse_barreb3_court.mat'	'rpsslackg3_1106_monophase_lignec_court.mat'
'rpsslackg3_1077_tripahse_barreb4_long.mat'	'rpsslackg3_1107_monophase_ligned_long.mat'
'rpsslackg3_1078_tripahse_barreb4_court.mat'	'rpsslackg3_1108_monophase_ligned_court.mat'
'rpsslackg3_1079_tripahse_barreb5_long.mat'	'rpsslackg3_1109_monophase_ligne_f_long.mat'
'rpsslackg3_1080_tripahse_barreb5_court.mat'	'rpsslackg3_1110_monophase_ligne_f_court.mat'
'rpsslackg3_1081_biphase_lignea_long.mat'	'rpsslackg3_1111_monophase_barreb1_long.mat'
'rpsslackg3_1082_biphase_lignea_court.mat'	'rpsslackg3_1112_monophase_barreb1_court.mat'
'rpsslackg3_1083_biphase_ligne_b_long.mat'	'rpsslackg3_1113_monophase_barreb2_long.mat'
'rpsslackg3_1084_biphase_ligne_b_court.mat'	'rpsslackg3_1114_monophase_barreb2_court.mat'
'rpsslackg3_1085_biphase_lignec_long.mat'	'rpsslackg3_1115_monophase_barreb3_long.mat'
'rpsslackg3_1086_biphase_lignec_court.mat'	'rpsslackg3_1116_monophase_barreb3_court.mat'
'rpsslackg3_1087_biphase_ligned_long.mat'	'rpsslackg3_1117_monophase_barreb4_long.mat'
'rpsslackg3_1088_biphase_ligned_court.mat'	'rpsslackg3_1118_monophase_barreb4_court.mat'
'rpsslackg3_1089_biphase_ligne_f_long.mat'	'rpsslackg3_1119_monophase_barreb5_long.mat'
'rpsslackg3_1090_biphase_ligne_f_court.mat'	'rpsslackg3_1120_monophase_barreb5_court.mat'

Noms des simulations lorsque le générateur 4 est défini comme Slack.

'rpsslackg4_1121_tripahse_lignea_long.mat'	'rpsslackg4_1151_biphase_barreb1_long.mat'
'rpsslackg4_1122_tripahse_lignea_court.mat'	'rpsslackg4_1152_biphase_barreb1_court.mat'
'rpsslackg4_1123_tripahse_ligneb_long.mat'	'rpsslackg4_1153_biphase_barreb2_long.mat'
'rpsslackg4_1124_tripahse_ligneb_court.mat'	'rpsslackg4_1154_biphase_barreb2_court.mat'
'rpsslackg4_1125_tripahse_lignec_long.mat'	'rpsslackg4_1155_biphase_barreb3_long.mat'
'rpsslackg4_1126_tripahse_lignec_court.mat'	'rpsslackg4_1156_biphase_barreb3_court.mat'
'rpsslackg4_1127_tripahse_ligned_long.mat'	'rpsslackg4_1157_biphase_barreb4_long.mat'
'rpsslackg4_1128_tripahse_ligned_court.mat'	'rpsslackg4_1158_biphase_barreb4_court.mat'
'rpsslackg4_1129_tripahse_lignef_long.mat'	'rpsslackg4_1159_biphase_barreb5_long.mat'
'rpsslackg4_1130_tripahse_lignef_court.mat'	'rpsslackg4_1160_biphase_barreb5_court.mat'
'rpsslackg4_1131_tripahse_barreb1_long.mat'	'rpsslackg4_1161_monophase_lignea_long.mat'
'rpsslackg4_1132_tripahse_barreb1_court.mat'	'rpsslackg4_1162_monophase_lignea_court.mat'
'rpsslackg4_1133_tripahse_barreb2_long.mat'	'rpsslackg4_1163_monophase_ligneb_long.mat'
'rpsslackg4_1134_tripahse_barreb2_court.mat'	'rpsslackg4_1164_monophase_ligneb_court.mat'
'rpsslackg4_1135_tripahse_barreb3_long.mat'	'rpsslackg4_1165_monophase_lignec_long.mat'
'rpsslackg4_1136_tripahse_barreb3_court.mat'	'rpsslackg4_1166_monophase_lignec_court.mat'
'rpsslackg4_1137_tripahse_barreb4_long.mat'	'rpsslackg4_1167_monophase_ligned_long.mat'
'rpsslackg4_1138_tripahse_barreb4_court.mat'	'rpsslackg4_1168_monophase_ligned_court.mat'
'rpsslackg4_1139_tripahse_barreb5_long.mat'	'rpsslackg4_1169_monophase_lignef_long.mat'
'rpsslackg4_1140_tripahse_barreb5_court.mat'	'rpsslackg4_1170_monophase_lignef_court.mat'
'rpsslackg4_1141_biphase_lignea_long.mat'	'rpsslackg4_1171_monophase_barreb1_long.mat'
'rpsslackg4_1142_biphase_lignea_court.mat'	'rpsslackg4_1172_monophase_barreb1_court.mat'
'rpsslackg4_1143_biphase_ligneb_long.mat'	'rpsslackg4_1173_monophase_barreb2_long.mat'
'rpsslackg4_1144_biphase_ligneb_court.mat'	'rpsslackg4_1174_monophase_barreb2_court.mat'
'rpsslackg4_1145_biphase_lignec_long.mat'	'rpsslackg4_1175_monophase_barreb3_long.mat'
'rpsslackg4_1146_biphase_lignec_court.mat'	'rpsslackg4_1176_monophase_barreb3_court.mat'
'rpsslackg4_1147_biphase_ligned_long.mat'	'rpsslackg4_1177_monophase_barreb4_long.mat'
'rpsslackg4_1148_biphase_ligned_court.mat'	'rpsslackg4_1178_monophase_barreb4_court.mat'
'rpsslackg4_1149_biphase_lignef_long.mat'	'rpsslackg4_1179_monophase_barreb5_long.mat'
'rpsslackg4_1150_biphase_lignef_court.mat'	'rpsslackg4_1180_monophase_barreb5_court.mat'

Annexe 2

Générateur 1						
N° de simulation pour :		Apprentissage		Test	Validation	
1001	1054	1099	1148	1002	1004	1118
1003	1055	1101	1150	1005	1008	1122
1006	1056	1105	1152	1010	1009	1127
1007	1057	1106	1153	1015	1016	1134
1012	1058	1107	1154	1026	1023	1140
1013	1061	1108	1155	1032	1024	1141
1014	1062	1110	1157	1036	1029	1143
1017	1063	1111	1158	1043	1030	1144
1018	1064	1112	1159	1053	1033	1151
1019	1065	1114	1160	1059	1038	1156
1020	1072	1120	1162	1067	1039	1161
1021	1074	1121	1163	1068	1042	1166
1022	1075	1123	1164	1071	1045	1167
1025	1076	1124	1165	1081	1047	1173
1027	1077	1125	1168	1095	1060	1177
1028	1078	1126	1169	1097	1066	1179
1031	1080	1128	1170	1100	1069	
1034	1082	1129	1171	1103	1070	
1035	1083	1130	1172	1115	1073	
1037	1085	1131	1174	1116	1079	
1040	1086	1132	1175	1117	1084	
1041	1087	1135	1176	1119	1089	
1044	1088	1136	1180	1133	1096	
1046	1090	1137		1138	1098	
1048	1091	1139		1142	1102	
1049	1092	1145		1149	1104	
1050	1093	1146		1178	1109	
1052	1094	1147			1113	

Générateur 2						
N° de simulation pour :		Apprentissage		Test	Validation	
1003	1053	1099	1146	1007	1001	1108
1004	1054	1100	1148	1009	1002	1113
1005	1055	1106	1150	1022	1006	1120
1008	1057	1107	1151	1032	1012	1121
1010	1058	1109	1152	1034	1013	1123
1014	1062	1110	1153	1036	1016	1125
1015	1063	1111	1154	1048	1021	1140
1017	1064	1112	1155	1059	1028	1142
1018	1066	1114	1160	1061	1029	1145
1019	1068	1115	1162	1065	1031	1149
1020	1070	1116	1163	1071	1038	1157
1023	1073	1117	1164	1078	1039	1158
1024	1074	1119	1165	1093	1045	1159
1025	1076	1124	1166	1094	1050	1161
1026	1079	1126	1168	1096	1052	1167
1027	1080	1127	1171	1101	1056	1170
1030	1082	1128	1172	1104	1060	1178
1033	1083	1129	1173	1118	1067	
1035	1084	1132	1175	1122	1069	
1037	1085	1134	1176	1130	1072	
1040	1086	1135	1177	1131	1075	
1041	1087	1136	1179	1133	1077	
1042	1089	1137	1180	1147	1081	
1043	1090	1138		1156	1088	
1044	1091	1139		1169	1092	
1046	1095	1141		1174	1102	
1047	1097	1143			1103	
1049	1098	1144			1105	

Générateur 3						
N° de simulation pour :		Apprentissage		Test	Validation	
1001	1046	1091	1147	1015	1006	1117
1002	1047	1092	1148	1024	1010	1119
1003	1049	1094	1149	1027	1013	1121
1004	1052	1097	1151	1030	1017	1122
1005	1053	1099	1152	1034	1018	1130
1007	1054	1102	1155	1040	1025	1132
1008	1055	1103	1156	1048	1031	1136
1009	1056	1105	1157	1062	1035	1140
1012	1058	1111	1158	1070	1039	1143
1014	1059	1112	1160	1079	1043	1144
1016	1060	1113	1162	1087	1050	1145
1019	1063	1114	1164	1100	1057	1154
1020	1065	1116	1165	1101	1061	1161
1021	1066	1118	1166	1104	1064	1167
1022	1067	1120	1169	1107	1071	1168
1023	1068	1124	1170	1109	1072	1172
1026	1069	1125	1171	1110	1080	1173
1028	1073	1126	1174	1123	1081	
1029	1074	1127	1175	1128	1084	
1032	1075	1133	1176	1129	1086	
1033	1076	1134	1177	1131	1088	
1036	1077	1135	1178	1150	1093	
1037	1078	1137	1179	1153	1095	
1038	1082	1138		1159	1096	
1041	1083	1139		1163	1098	
1042	1085	1141		1180	1106	
1044	1089	1142			1108	
1045	1090	1146			1115	

Générateur 4						
N° de simulation pour :		Apprentissage		Test	Validation	
1002	1056	1104	1150	1008	1001	1102
1003	1059	1105	1151	1016	1007	1108
1004	1060	1106	1152	1017	1012	1117
1005	1062	1107	1153	1024	1015	1132
1006	1064	1109	1154	1028	1022	1133
1009	1065	1111	1155	1029	1023	1134
1010	1070	1113	1157	1043	1034	1137
1013	1072	1114	1158	1044	1035	1139
1014	1075	1115	1159	1049	1040	1140
1018	1076	1116	1160	1052	1041	1144
1019	1077	1118	1161	1054	1042	1149
1020	1079	1120	1162	1057	1046	1156
1021	1080	1121	1163	1066	1047	1169
1025	1081	1123	1164	1071	1048	1170
1026	1083	1124	1165	1073	1058	1171
1027	1085	1125	1166	1100	1061	1175
1030	1086	1126	1167	1110	1063	
1031	1087	1128	1168	1112	1067	
1032	1088	1129	1173	1119	1068	
1033	1089	1130	1176	1122	1069	
1036	1092	1131	1178	1127	1074	
1037	1094	1135	1179	1143	1078	
1038	1095	1136	1180	1146	1082	
1039	1096	1138		1147	1084	
1045	1097	1141		1172	1090	
1050	1098	1142		1174	1091	
1053	1101	1145		1177	1093	
1055	1103	1148			1099	

Annexe 3

```

prepare_parametric_case();

var Current = currentCircuit();

// Appelle la fonction qui initialise le circuit de base, ferme les
// Disjoncteur de ligne et ouvre les disjoncteur de court-circuit
Initialisation();

i=125;                // prévient le lancement par erreur
                        // Si i = 9999, les 60 simulations sont lancées
setSimul=120; //numéro initial au milieu du nom de la simulation
                        // 1 pour RPSSlackG1
                        // 60 pour RPSSlackG3
                        // 120 pour RPSSlackG4

/////////////////////////
/////////////////////////

// Debut des court circuit Triphase

/////////////////////////
/////////////////////////

if(i==9999){
// Simulation 1
Initialisation();
k=1001+setSimul; // Numéro de simulation dans le nom
name = ' '+k+'_Tripahse_LigneA_Long'; // Définis le nom de la simulation
// Ferme l'interrupteur MAT_a_G à 1000ms et l'ouvre à 1200ms
SwitchTri('1000ms','1000ms','1000ms','1200ms','1200ms','1200ms','MAT_a_G');
// Lance une simulation temporelle, et sauvegarde la simulation avec le nom "name"
run_sim(name);

// Simulation 2
Initialisation();
k=1002+setSimul;
name = ' '+k+'_Tripahse_LigneA_Court';
SwitchTri('1000ms','1000ms','1000ms','1100ms','1100ms','1100ms','MAT_a_G');
run_sim(name);

// Simulation 3
Initialisation();
k=1003+setSimul;
name = ' '+k+'_Tripahse_LigneB_Long';
SwitchTri('1000ms','1000ms','1000ms','1200ms','1200ms','1200ms','MAT_b_G');
run_sim(name);

// Simulation 4

```

```

Initialisation();
k=1004+setSimul;
name = '_'+k+'_Tripahse_LigneB_Court';
SwitchTri('1000ms','1000ms','1000ms','1100ms','1100ms','1100ms','MAT_b_G');
run_sim(name);

// Simulation 5
Initialisation();
k=1005+setSimul;
name = '_'+k+'_Tripahse_LigneC_Long';
SwitchTri('1000ms','1000ms','1000ms','1200ms','1200ms','1200ms','MAT_c_G');
run_sim(name);

// Simulation 6
Initialisation();
k=1006+setSimul;
name = '_'+k+'_Tripahse_LigneC_Court';
SwitchTri('1000ms','1000ms','1000ms','1100ms','1100ms','1100ms','MAT_c_G');
run_sim(name);

// Simulation 7
Initialisation();
k=1007+setSimul;
name = '_'+k+'_Tripahse_LigneD_Long';
SwitchTri('1000ms','1000ms','1000ms','1200ms','1200ms','1200ms','MAT_d_G');
run_sim(name);

// Simulation 8
Initialisation();
k=1008+setSimul;
name = '_'+k+'_Tripahse_LigneD_Court';
SwitchTri('1000ms','1000ms','1000ms','1100ms','1100ms','1100ms','MAT_d_G');
run_sim(name);

// Simulation 9
Initialisation();
k=1009+setSimul;
name = '_'+k+'_Tripahse_LigneF_Long';
SwitchTri('1000ms','1000ms','1000ms','1200ms','1200ms','1200ms','MAT_f_G');
run_sim(name);

// Simulation 10
Initialisation();
k=1010+setSimul;
name = '_'+k+'_Tripahse_LigneF_Court';
SwitchTri('1000ms','1000ms','1000ms','1100ms','1100ms','1100ms','MAT_f_G');
run_sim(name);

// Simulation 11
Initialisation();
k=1011+setSimul;
name = '_'+k+'_Tripahse_BarreB1_Long';
SwitchTri('1000ms','1000ms','1000ms','1200ms','1200ms','1200ms','MAT_B1');
run_sim(name);

// Simulation 12
Initialisation();
k=1012+setSimul;
name = '_'+k+'_Tripahse_BarreB1_Court';
SwitchTri('1000ms','1000ms','1000ms','1100ms','1100ms','1100ms','MAT_B1');
run_sim(name);

```

```
// Simulation 13
Initialisation();
k=1013+setSimul;
name = '_'+k+'_Tripahse_BarreB2_Long';
SwitchTri('1000ms','1000ms','1000ms','1200ms','1200ms','1200ms','MAT_B2');
run_sim(name);
```

```
// Simulation 14
Initialisation();
k=1014+setSimul;
name = '_'+k+'_Tripahse_BarreB2_Court';
SwitchTri('1000ms','1000ms','1000ms','1100ms','1100ms','1100ms','MAT_B2');
run_sim(name);
```

```
// Simulation 15
Initialisation();
k=1015+setSimul;
name = '_'+k+'_Tripahse_BarreB3_Long';
SwitchTri('1000ms','1000ms','1000ms','1200ms','1200ms','1200ms','MAT_B3');
run_sim(name);
```

```
// Simulation 16
Initialisation();
k=1016+setSimul;
name = '_'+k+'_Tripahse_BarreB3_Court';
SwitchTri('1000ms','1000ms','1000ms','1100ms','1100ms','1100ms','MAT_B3');
run_sim(name);
```

```
// Simulation 17
Initialisation();
k=1017+setSimul;
name = '_'+k+'_Tripahse_BarreB4_Long';
SwitchTri('1000ms','1000ms','1000ms','1200ms','1200ms','1200ms','MAT_B4');
run_sim(name);
```

```
// Simulation 18
Initialisation();
k=1018+setSimul;
name = '_'+k+'_Tripahse_BarreB4_Court';
SwitchTri('1000ms','1000ms','1000ms','1100ms','1100ms','1100ms','MAT_B4');
run_sim(name);
```

```
// Simulation 19
Initialisation();
k=1019+setSimul;
name = '_'+k+'_Tripahse_BarreB5_Long';
SwitchTri('1000ms','1000ms','1000ms','1200ms','1200ms','1200ms','MAT_B5');
run_sim(name);
```

```
// Simulation 20
Initialisation();
k=1020+setSimul;
name = '_'+k+'_Tripahse_BarreB5_Court';
SwitchTri('1000ms','1000ms','1000ms','1100ms','1100ms','1100ms','MAT_B5');
run_sim(name);
```

```
////////////////////////////////////
////////////////////////////////////
```

```
// Debut des court circuit biphasé
```

```
////////////////////////////////////
```

```

////////////////////////////////////

// Simulation 21
Initialisation();
k=1021+setSimul;
name = '_'+k+'_Biphase_LigneA_Long'; // Règle le nom de la simulation
// Ferme l'interrupteur CC_a_AB à 1000ms et l'ouvre à 1200ms
SwitchMono('1000ms','1200ms','CC_a_AB');
run_sim(name);

// Simulation 22
Initialisation();
k=1022+setSimul;
name = '_'+k+'_Biphase_LigneA_Court';
SwitchMono('1000ms','1100ms','CC_a_AB');
run_sim(name);

// Simulation 23
Initialisation();
k=1023+setSimul;
name = '_'+k+'_Biphase_LigneB_Long';
SwitchMono('1000ms','1200ms','CC_b_AB');
run_sim(name);

// Simulation 24
Initialisation();
k=1024+setSimul;
name = '_'+k+'_Biphase_LigneB_Court';
SwitchMono('1000ms','1100ms','CC_b_AB');
run_sim(name);

// Simulation 25
Initialisation();
k=1025+setSimul;
name = '_'+k+'_Biphase_LigneC_Long';
SwitchMono('1000ms','1200ms','CC_c_AB');
run_sim(name);

// Simulation 26
Initialisation();
k=1026+setSimul;
name = '_'+k+'_Biphase_LigneC_Court';
SwitchMono('1000ms','1100ms','CC_c_AB');
run_sim(name);

// Simulation 27
Initialisation();
k=1027+setSimul;
name = '_'+k+'_Biphase_LigneD_Long';
SwitchMono('1000ms','1200ms','CC_d_AB');
run_sim(name);

// Simulation 28
Initialisation();
k=1028+setSimul;
name = '_'+k+'_Biphase_LigneD_Court';
SwitchMono('1000ms','1100ms','CC_d_AB');
run_sim(name);

// Simulation 29

```

```

Initialisation();
k=1029+setSimul;
name = ' '+k+'_Biphase_LigneF_Long';
SwitchMono('1000ms','1200ms','CC_f_AB');
run_sim(name);

// Simulation 30
Initialisation();
k=1030+setSimul;
name = ' '+k+'_Biphase_LigneF_Court';
SwitchMono('1000ms','1100ms','CC_f_AB');
run_sim(name);

// Simulation 31
Initialisation();
k=1031+setSimul;
name = ' '+k+'_Biphase_BarreB1_Long';
SwitchMono('1000ms','1200ms','CC_B1_AB');
run_sim(name);

// Simulation 32
Initialisation();
k=1032+setSimul;
name = ' '+k+'_Biphase_BarreB1_Court';
SwitchMono('1000ms','1100ms','CC_B1_AB');
run_sim(name);

// Simulation 33
Initialisation();
k=1033+setSimul;
name = ' '+k+'_Biphase_BarreB2_Long';
SwitchMono('1000ms','1200ms','CC_B2_AB');
run_sim(name);

// Simulation 34
Initialisation();
k=1034+setSimul;
name = ' '+k+'_Biphase_BarreB2_Court';
SwitchMono('1000ms','1100ms','CC_B2_AB');
run_sim(name);

// Simulation 35
Initialisation();
k=1035+setSimul;
name = ' '+k+'_Biphase_BarreB3_Long';
SwitchMono('1000ms','1200ms','CC_B3_AB');
run_sim(name);

// Simulation 36
Initialisation();
k=1036+setSimul;
name = ' '+k+'_Biphase_BarreB3_Court';
SwitchMono('1000ms','1100ms','CC_B3_AB');
run_sim(name);

// Simulation 37
Initialisation();
k=1037+setSimul;
name = ' '+k+'_Biphase_BarreB4_Long';
SwitchMono('1000ms','1200ms','CC_B4_AB');
run_sim(name);

// Simulation 38

```

```

Initialisation();
k=1038+setSimul;
name = ' '+k+' Biphase_BarreB4_Court';
SwitchMono('1000ms','1100ms','CC_B4_AB');
run_sim(name);

// Simulation 39
Initialisation();
k=1039+setSimul;
name = ' '+k+' Biphase_BarreB5_Long';
SwitchMono('1000ms','1200ms','CC_B5_AB');
run_sim(name);

// Simulation 40
Initialisation();
k=1040+setSimul;
name = ' '+k+' Biphase_BarreB5_Court';
SwitchMono('1000ms','1100ms','CC_B5_AB');
run_sim(name);

////////////////////////////////////
////////////////////////////////////

// Debut des court circuit Monophase

////////////////////////////////////
////////////////////////////////////

// Simulation 41
Initialisation();
k=1041+setSimul;
name = ' '+k+' Monophase_LigneA_Long';
//
SwitchTri('1000ms','500s','500s','1200ms','500s','500s','MAT_a_G');

run_sim(name);

// Simulation 42
Initialisation();
k=1042+setSimul;
name = ' '+k+' Monophase_LigneA_Court';
SwitchTri('1000ms','500s','500s','1100ms','500s','500s','MAT_a_G');

run_sim(name);

// Simulation 43
Initialisation();
k=1043+setSimul;
name = ' '+k+' Monophase_LigneB_Long';
SwitchTri('1000ms','500s','500s','1200ms','500s','500s','MAT_b_G');

run_sim(name);

// Simulation 44
Initialisation();
k=1044+setSimul;
name = ' '+k+' Monophase_LigneB_Court';
SwitchTri('1000ms','500s','500s','1100ms','500s','500s','MAT_b_G');

run_sim(name);

// Simulation 45
Initialisation();
k=1045+setSimul;

```

```

name = ' '+k+' Monophase_LigneC_Long';
SwitchTri('1000ms','500s','500s','1200ms','500s','500s','MAT_c_G');

run_sim(name);

// Simulation 46
Initialisation();
k=1046+setSimul;
name = ' '+k+' Monophase_LigneC_Court';
SwitchTri('1000ms','500s','500s','1100ms','500s','500s','MAT_c_G');

run_sim(name);

// Simulation 47
Initialisation();
k=1047+setSimul;
name = ' '+k+' Monophase_LigneD_Long';
SwitchTri('1000ms','500s','500s','1200ms','500s','500s','MAT_d_G');

run_sim(name);

// Simulation 48
Initialisation();
k=1048+setSimul;
name = ' '+k+' Monophase_LigneD_Court';
SwitchTri('1000ms','500s','500s','1100ms','500s','500s','MAT_d_G');

run_sim(name);

// Simulation 49
Initialisation();
k=1049+setSimul;
name = ' '+k+' Monophase_LigneF_Long';
SwitchTri('1000ms','500s','500s','1200ms','500s','500s','MAT_f_G');

run_sim(name);

// Simulation 50
Initialisation();
k=1050+setSimul;
name = ' '+k+' Monophase_LigneF_Court';
SwitchTri('1000ms','500s','500s','1100ms','500s','500s','MAT_f_G');
run_sim(name);

// Simulation 51
Initialisation();
k=1051+setSimul;
name = ' '+k+' Monophase_BarreB1_Long';
SwitchTri('1000ms','500s','500s','1200ms','500s','500s','MAT_B1');
run_sim(name);

// Simulation 52
Initialisation();
k=1052+setSimul;
name = ' '+k+' Monophase_BarreB1_Court';
SwitchTri('1000ms','500s','500s','1100ms','500s','500s','MAT_B1');
run_sim(name);

// Simulation 53
Initialisation();
k=1053+setSimul;
name = ' '+k+' Monophase_BarreB2_Long';
SwitchTri('1000ms','500s','500s','1200ms','500s','500s','MAT_B2');

```

```

run_sim(name);

// Simulation 54
Initialisation();
k=1054+setSimul;
name = '_'+k+'_Monophase_BarreB2_Court';
SwitchTri('1000ms','500s','500s','1100ms','500s','500s','MAT_B2');
run_sim(name);

// Simulation 55
Initialisation();
k=1055+setSimul;
name = '_'+k+'_Monophase_BarreB3_Long';
SwitchTri('1000ms','500s','500s','1200ms','500s','500s','MAT_B3');
run_sim(name);

// Simulation 56
Initialisation();
k=1056+setSimul;
name = '_'+k+'_Monophase_BarreB3_Court';
SwitchTri('1000ms','500s','500s','1100ms','500s','500s','MAT_B3');
run_sim(name);

// Simulation 57
Initialisation();
k=1057+setSimul;
name = '_'+k+'_Monophase_BarreB4_Long';
SwitchTri('1000ms','500s','500s','1200ms','500s','500s','MAT_B4');
run_sim(name);

// Simulation 58
Initialisation();
k=1058+setSimul;
name = '_'+k+'_Monophase_BarreB4_Court';
SwitchTri('1000ms','500s','500s','1100ms','500s','500s','MAT_B4');
run_sim(name);

// Simulation 59
Initialisation();
k=1059+setSimul;
name = '_'+k+'_Monophase_BarreB5_Long';
SwitchTri('1000ms','500s','500s','1200ms','500s','500s','MAT_B5');
run_sim(name);

// Simulation 60
Initialisation();
k=1060+setSimul;
name = '_'+k+'_Monophase_BarreB5_Court';
SwitchTri('1000ms','500s','500s','1100ms','500s','500s','MAT_B5');
run_sim(name);
}

// Créé par EMTP -- Initialise le dossier de simulation et prépare le noms de fichiers
function prepare_parametric_case(){
    /*Some useful script functions
    parseScriptFile('make_file_name.dwj');
    parseScriptFile('program_dwj');
    parseScriptFile('run_emtp.dwj');
    parseScriptFile('check_my_filename.dwj');
    parseScriptFile('find_data_in_object.dwj'); //for searching objects

    /*Prepare design_folder
    prepare_design_folder();
}

// Lance une simulation sauvegarde le nom du fichier en tant que nom de PLOT

```

```

function run_sim(nomSimul,k){
    writeln(nomSimul +'          _____Debut de la simulation_____');
    str=runExportToString('report scripts/EMTP Netlist.rfm',null,'NETLIST');
    var netlist_file = Netlist_file_name();
    run_emtp(netlist_file,false,'1');
    rename_plot_file(nomSimul,Current);
    writeln(nomSimul +'          **** Simulation Finie ****');
}

// Fonction a appeler pour modifier la compensation de totues les lignes à 5%
function comp5()
{
    Comp(0.00912,0.00912,0.00912,'CS_a_D');
    Comp(0.00912,0.00912,0.00912,'CS_b_D');
    Comp(0.0076,0.0076,0.0076,'CS_c_D');
    Comp(0.0076,0.0076,0.0076,'CS_d_D');
    Comp(0.0076,0.0076,0.0076,'CS_e_D');
    Comp(0.0076,0.0076,0.0076,'CS_f_D');
    Comp(0.0076,0.0076,0.0076,'CS_g_D');
}

// Fonction a appeler pour modifier la compensation de totues les lignes à 30%
function comp30()
{
    Comp(0.1824,0.1824,0.1824,'CS_a_D');
    Comp(0.1824,0.1824,0.1824,'CS_b_D');
    Comp(0.152,0.152,0.152,'CS_c_D');
    Comp(0.152,0.152,0.152,'CS_d_D');
    Comp(0.152,0.152,0.152,'CS_e_D');
    Comp(0.152,0.152,0.152,'CS_f_D');
    Comp(0.152,0.152,0.152,'CS_g_D');
}

// Fonction qui va changer le temps d'ouverture et de fermeture
// du SWITCH monophasé passé en paramètre par le nom "Name"
function SwitchMono(TopenA,TcloseA,Name){
    var cct = currentCircuit();
    dev=cct.devices(4,-1,3,"Name", Name);
    dev=dev[0];
    DataA=dev.getAttribute('ParamsA');
    DataA=DataA.split(",");
    DataA[0]=TopenA;
    DataA[1]=TcloseA;
    dev.setAttribute('ParamsA',DataA+',');
}

// Fonction qui va changer le temps d'ouverture et de fermeture
// du SWITCH triphasé passé en paramètre par le nom "Name"
function SwitchTri(TopenA,TopenB,TopenC,TcloseA,TcloseB,TcloseC,Name){
    var cct = currentCircuit();
    dev=cct.devices(4,-1,3,"Name", Name);
    dev=dev[0];
    DataA=dev.getAttribute('ParamsA');
    DataA=DataA.split(",");
    DataA[0]=TopenA;
    DataA[1]=TcloseA;
    dev.setAttribute('ParamsA',DataA+',');
    DataB=dev.getAttribute('ParamsB');
    DataB=DataB.split(',');
    DataB[0]=TopenB;
    DataB[1]=TcloseB;
    dev.setAttribute('ParamsB',DataB+',');
    DataC=dev.getAttribute('ParamsC');
    DataC=DataC.split(',');
    DataC[0]=TopenC;
    DataC[1]=TcloseC;
    dev.setAttribute('ParamsC',DataC+',');
}

```

```

}

// Fonction qui modifie la longueur de la ligne passée en paramètre
function LongueurLigne(Long,Nom){

    var cct = currentCircuit();
    dev=cct.devices(4,-1,3,"Name", Nom);
    dev=dev[0];
    Data=dev.getAttribute('ModelData');
    Data=Data.split('\n');
    Data[0]=Data[0].split(' ');
    Data[1]=Data[1].split(' ');
    Data[0][0]=Long;
    Data[1][0]=Long;
    Model=Data[0][0]+"      "+Data[0][1]+"      "+Data[0][2]+"      "+Data[0][3]+" \n"+Data[1][0]+"      "+Data[1][1]+"      "+Data[1][2]+"
    "+Data[1][3]+" \n"+0+" \n"+0+" \n"+0;
    dev.setAttribute('ModelData',Model);
}

// Fonction qui va changer individuellement la valeur de la capacité passé en paramètre
function Comp(CompA,CompB,CompC,Name){
    var cct = currentCircuit();
    dev=cct.devices(4,-1,3,"Name", Name);
    dev=dev[0];
    DataA=dev.getAttribute('ParamsA');
    DataA=DataA.split(",");
    DataA[2]=CompA;
    dev.setAttribute('ParamsA',DataA+',');
    DataB=dev.getAttribute('ParamsB');
    DataB=DataB.split(',');
    DataB[2]=CompB;
    dev.setAttribute('ParamsB',DataB+',');
    DataC=dev.getAttribute('ParamsC');
    DataC=DataC.split(',');
    DataC[2]=CompC;
    dev.setAttribute('ParamsC',DataC+',');
}

// FOnction qui cange les paramètre des imédances de terre
function ImpGr(Ra,La,Ca,Rb,Lb,Cb,Rc,Lc,Cc,Name){
    var cct = currentCircuit();
    dev=cct.devices(4,-1,3,"Name", Name);
    dev=dev[0];
    DataA=dev.getAttribute('ParamsA');
    DataA=DataA.split(",");
    DataA[0]=Ra;
    DataA[1]=La;
    DataA[2]=Ca;
    dev.setAttribute('ParamsA',DataA+',');
    DataB=dev.getAttribute('ParamsB');
    DataB=DataB.split(',');
    DataB[0]=Rb;
    DataB[1]=Lb;
    DataB[2]=Cb;
    dev.setAttribute('ParamsB',DataB+',');
    DataC=dev.getAttribute('ParamsC');
    DataC=DataC.split(',');
    DataC[0]=Rc;
    DataC[1]=Lc;
    DataC[2]=Cc;
    dev.setAttribute('ParamsC',DataC+',');
}

// FOnction d'initialisation du système anderson de base
function Initialisation(){ // Initialisation des fermeture des disjoncteurs

```

```

for(i=97;i<=103;i++){

    Switch='Ouv_'+String.fromCharCode(i)+'_D';
    SwitchTri(-1,-1,-1,500,500,500,Switch);
    Switch='Ouv_'+String.fromCharCode(i)+'_G';
    SwitchTri(-1,-1,-1,500,500,500,Switch);

    Switch='Disj_'+String.fromCharCode(i)+'_D';
    SwitchTri(-1,-1,-1,500,500,500,Switch);
    Switch='Disj_'+String.fromCharCode(i)+'_G';
    SwitchTri(-1,-1,-1,500,500,500,Switch);

    // Switch Triphasé de mise à la terre initialement ouvert et ouvert tout le long de la simulation
    Switch='MAT_'+String.fromCharCode(i)+'_D';
    SwitchTri(500,500,500,500,500,500,Switch);
    Switch='MAT_'+String.fromCharCode(i)+'_G';
    SwitchTri(500,500,500,500,500,500,Switch);

    // Switch entrephases initialement ouvert et jamais ouvert tout le long de la simulation
    Switch='CC_'+String.fromCharCode(i)+'_AB';
    SwitchMono(500,0,Switch);
    Switch='CC_'+String.fromCharCode(i)+'_BC';
    SwitchMono(500,0,Switch);
    Switch='CC_'+String.fromCharCode(i)+'_AC';
    SwitchMono(500,0,Switch);

    // Mise a 0 des composants RLC de terre
    ImpeRLC='RLC_'+String.fromCharCode(i)+'_G';
    ImpGr(0,0,0,0,0,0,0,0,ImpeRLC);
    ImpeRLC='RLC_'+String.fromCharCode(i)+'_D';
    ImpGr(0,0,0,0,0,0,0,0,ImpeRLC);
}

// Initialisation des disjoncteurs de generateurs, initialement fermé jamais ouvert
Switch='Disj_Gen1';
SwitchTri(-1,-1,-1,500,500,500,Switch);
Switch='Disj_Gen2';
SwitchTri(-1,-1,-1,500,500,500,Switch);
Switch='Disj_Gen3';
SwitchTri(-1,-1,-1,500,500,500,Switch);
Switch='Disj_Gen4';
SwitchTri(-1,-1,-1,500,500,500,Switch);

// Initialisation des lignes et compensateur séries
for(i=97;i<=98;i++){
    Ligne='Ligne_'+String.fromCharCode(i)+'_G';
    LongueurLigne(300,Ligne);
    Ligne='Ligne_'+String.fromCharCode(i)+'_D';
    LongueurLigne(300,Ligne);
    CompStat='CS_'+String.fromCharCode(i)+'_D';
    Comp(0.1824,0.1824,0.1824,CompStat);
}
for(i=99;i<=103;i++){
    Ligne='Ligne_'+String.fromCharCode(i)+'_G';
    LongueurLigne(250,Ligne);
    Ligne='Ligne_'+String.fromCharCode(i)+'_D';
    LongueurLigne(250,Ligne);
    CompStat='CS_'+String.fromCharCode(i)+'_D';
    Comp(0.152,0.152,0.152,CompStat);
}
for(k=1;k<=6;k++){
    Switch='MAT_B'+k;
    SwitchTri(500,500,500,500,500,500,Switch);
    Switch='CC_B'+k+'_AB';
    SwitchMono(500,500,Switch);
}
}

```

Annexe 4

Code MATLAB pour l'importation des données, et de la séparation des matrices de chaque machines.

```
%% Importation des donnees;

% Ouvre chaque fichier, transforme les données en Single,
% supprime tous les élément du vecteur électrique supérieur à 5 secondes
% et crée le vecteur temps

% Répertoire les simulations et calcul le nombre

FichierSimul = dir('*.mat');
NombreDeSimul = length(FichierSimul);

for k = 1:NombreDeSimul
    nom=FichierSimul(k,1).name;
    load(nom);
    [long larg]=size(Ylabels);
    t=cell2mat(X(1));
    y=cell2mat(Y);
    clear Xlabels X Y VectMec
    y = single(downsample(y,1));
    t = single(downsample(t,1));

    % Cherche la position dans la matrice des différentes variables
    % mécaniques et crée la matrice VectMec
    for l = 1:larg
        if(strcmp(Ylabels(1,l),'PowerAng_Gen1@machine@1'))
            VectMec(:,1)=y(:,1);
        end
        if(strcmp(Ylabels(1,l),'PowerAng_Gen2@machine@1'))
            VectMec(:,2)=y(:,1);
        end
        if(strcmp(Ylabels(1,l),'PowerAng_Gen3@machine@1'))
            VectMec(:,3)=y(:,1);
        end
        if(strcmp(Ylabels(1,l),'PowerAng_Gen4@machine@1'))
            VectMec(:,4)=y(:,1);
        end
        if(strcmp(Ylabels(1,l),'Omega_1_Gen1@machine@1'))
            VectMec(:,5)=y(:,1);
        end
        if(strcmp(Ylabels(1,l),'Omega_1_Gen2@machine@1'))
            VectMec(:,6)=y(:,1);
        end
        if(strcmp(Ylabels(1,l),'Omega_1_Gen3@machine@1'))
            VectMec(:,7)=y(:,1);
        end
    end
end
```

```

        end
        if(strcmp(Ylabels(1,1),'Omega_1_Gen4@machine@1 '))
            VectMec(:,8)=y(:,1);
        end
    end

    % Détermination de l'individu correspondant à 5 secondes.
    i=1;
    while t(i)<5
        if i==length(t)
            break
        end
        i=i+1;
    end
    % Créé le vecteur électrique long de 5 secondes et les labels associés
    VectElect=y([1:i],[1:larg-8]);
    Ylabels(:,[157:164])=[];
    save('RPS.mat','t','VectMec','VectElect','Ylabels');
end

%% Separation de la base de donnee en 4

% Ce code sépare la matrice VectElect en VectElectGen1, VectElectGen2,
% VectElectGen3, VectElectGen4 avec leur Labels correspondant.
FichierSimul = dir('*.mat');
NombreDeSimul = length(FichierSimul);

for j = 1:NombreDeSimul
    nom=FichierSimul(j,1).name;
    load(nom);
    [long larg]=size(Ylabels);
    k=0;
    l=0;
    m=0;
    n=0;
    % cherche tous les éléments Gen1 Gen2 Gen3 Gen4 et classe dans leur
    % vecteur respectif les données électriques et leur labels
    for o = 1:larg
        if(isempty(strfind(Ylabels{1,o}, 'Gen1'))==0)
            k=k+1;
            VectElectGen1(:,k)=VectElect(:,o);
            LabelsGen1{1,k}=Ylabels{1,o};
        end
        if(isempty(strfind(Ylabels{1,o}, 'Gen2'))==0)
            l=l+1;
            VectElectGen2(:,l)=VectElect(:,o);
            LabelsGen2{1,l}=Ylabels{1,o};
        end
        if(isempty(strfind(Ylabels{1,o}, 'Gen3'))==0)
            m=m+1;
            VectElectGen3(:,m)=VectElect(:,o);
            LabelsGen3{1,m}=Ylabels{1,o};
        end
        if(isempty(strfind(Ylabels{1,o}, 'Gen4'))==0)
            n=n+1;
            VectElectGen4(:,n)=VectElect(:,o);
            LabelsGen4{1,n}=Ylabels{1,o};
        end
    end
    save(nom,'LabelsGen1','LabelsGen2','LabelsGen3','LabelsGen4'...
        ,'VectElectGen1','VectElectGen2','VectElectGen3'...
        ,'VectElectGen4','VectMec','t');
    clear Labels* Vect* t
end
clear all
clc

```

Annexe 5

Code MATLAB la détection des machines instables avec le COI et ajout de la matrice status ainsi que des différentes variables non mesurées par EMTP-RV

```
%% COI et COA

% Ce code permet de déterminer le centre d'inertie et d'angle de 4
% machines.

% Il permet aussi de visualiser la moyenne de l'angle et de la
% vitesse de dernier 480 points en fonction du numéro de simulation.

% Finalement, insère dans chaque simulation le vecteur status qui
% détermine la stabilité

% Liste toutes les simulations et en calcul le nombre
FichierSimul = dir('*.mat');
NombreDeSimul = length(FichierSimul);

% Définition du seuil, calculé selon le rapport
Seuil=0.0002;
for k = 1:NombreDeSimul

    nom=FichierSimul(k,1).name;
    load(nom)
    % Moments d'inertie des machines
    M1=4.99*10^6;
    M4=5.82*10^6;
    M2=84.38*10^3;
    M3=285.6*10^3;
    Mt=M1+M2+M3+M4;

    Delta(:, [1:4])=unwrap(VectMec(:, [1:4]));

    % Calcule l'angle moyen pondéré par le moment d'inertie des machines et
    % divisé par le moment total
    VectDeltaCOI=1/Mt*(Delta(:,1)*M1+Delta(:,2)*M2+Delta(:,3)*...
        M3+Delta(:,4)*M4);
    % Calcul le delta COI comme mentionné dans le rapport
    DeltaCOI(:,1)=Delta(:,1)-VectDeltaCOI;
    DeltaCOI(:,2)=Delta(:,2)-VectDeltaCOI;
    DeltaCOI(:,3)=Delta(:,3)-VectDeltaCOI;
    DeltaCOI(:,4)=Delta(:,4)-VectDeltaCOI;

    % Calcule l'angle moyen dans les 480 derniers points
    MeanDeltaCOI1=mean(DeltaCOI(end-480:end,1));
    MeanDeltaCOI2=mean(DeltaCOI(end-480:end,2));
```

```

MeanDeltaCOI3=mean(DeltaCOI(end-480:end,3));
MeanDeltaCOI4=mean(DeltaCOI(end-480:end,4));

% Calcule la vitesse moyenne pondéré par le moment d'inertie des
% machines et divisé par le moment total
OCOI=1/Mt*(M1*VectMec(:,5)+M2*VectMec(:,6)+...
    M3*VectMec(:,7)+M4*VectMec(:,8));

% Calcul le Omega COI comme mentionné dans le rapport
OM1COI=VectMec(:,5)-OCOI;
OM2COI=VectMec(:,6)-OCOI;
OM3COI=VectMec(:,7)-OCOI;
OM4COI=VectMec(:,8)-OCOI;

% Calcule la vitesse moyenne dans les 480 derniers points
MeanOmegaCOI1=mean(OM1COI(end-480:end,1));
MeanOmegaCOI2=mean(OM2COI(end-480:end,1));
MeanOmegaCOI3=mean(OM3COI(end-480:end,1));
MeanOmegaCOI4=mean(OM4COI(end-480:end,1));

% Initialisation du vecteur status
clear status
status=zeros(length(VectElectGen1),4);

% Détermine si la simulation est déclaré divergente ou convergente
if (abs(MeanOmegaCOI1)>(Seuil) && abs(MeanDeltaCOI1)>pi)

% Plot des simulations
figure(1)
plot(k,MeanDeltaCOI1,'bs',k,MeanDeltaCOI1,'bx')
hold on
figure(2)
plot(k,MeanOmegaCOI1,'bs',k,MeanOmegaCOI1,'bx')
hold on

    % Vérifie si le courtcircuit est long ou court et place 1 et
    % place 1 dans le vecteur status

    if(isempty(strfind(nom, 'long'))==0)
        status([4608 :end],1)=1;
    else
        status([4224 :end],1)=1;
    end
else % Entre si la simulation est stable

%
figure(1)
plot(k,MeanDeltaCOI1,'bs')
hold on
figure(2)
plot(k,MeanOmegaCOI1,'bs')
hold on

    % Vérifie si le courtcircuit est long ou court et place 1 et
    % placé 0 dans le vecteur status

    if(isempty(strfind(nom, 'long'))==0)
        status([4608 :end],1)=0;
    else
        status([4224 :end],1)=0;
    end
end

% Générateur 2
if (abs(MeanOmegaCOI2)>(Seuil) && abs(MeanDeltaCOI2)>pi)

%
figure(1)
plot(k,MeanDeltaCOI2,'gs',k,MeanDeltaCOI2,'gx')

```

```

% hold on
% figure(2)
% plot(k,MeanOmegaCOI2,'gs',k,MeanOmegaCOI2,'gx')
% hold on
%

        if(isempty(strfind(nom, 'long'))==0)
            status([4608 :end],2)=1;
        else
            status([4224 :end],2)=1;
        end

    else

%
% figure(1)
% plot(k,MeanDeltaCOI2,'gs')
% hold on
% figure(2)
% plot(k,MeanOmegaCOI2,'gs')
% hold on

        if(isempty(strfind(nom, 'long'))==0)
            status([4608 :end],2)=0;
        else
            status([4224 :end],2)=0;
        end

    end

% Générateur 3
if (abs(MeanOmegaCOI3)>(Seuil) && abs(MeanDeltaCOI3)>pi)

%
% figure(1)
% plot(k,MeanDeltaCOI3,'rs',k,MeanDeltaCOI3,'rx')
% hold on
% figure(2)
% plot(k,MeanOmegaCOI3,'rs',k,MeanOmegaCOI3,'rx')
% hold on

        if(isempty(strfind(nom, 'long'))==0)
            status([4608 :end],3)=1;
        else
            status([4224 :end],3)=1;
        end

    else

%
% figure(1)
% plot(k,MeanDeltaCOI3,'s')
% hold on
% figure(2)
% plot(k,MeanOmegaCOI3,'rs')
% hold on
% %

        if(isempty(strfind(nom, 'long'))==0)
            status([4608 :end],3)=0;
        else
            status([4224 :end],3)=0;
        end

    end

end

```

```

% Générateur 4
if (abs(MeanOmegaCOI4)>(Seuil) && abs(MeanDeltaCOI4)>pi)
%   figure(1)
%   plot(k,MeanDeltaCOI4,'cs',k,MeanDeltaCOI4,'cx')
%   hold on
%   figure(2)
%   plot(k,MeanOmegaCOI4,'cs',k,MeanOmegaCOI4,'cx')
%   hold on

        if(isempty(strfind(nom, 'long'))==0)
            status([4608 :end],4)=1;
        else
            status([4224 :end],4)=1;
        end

else
%
%   figure(1)
%   plot(k,MeanDeltaCOI4,'cs')
%   hold on
%   figure(2)
%   plot(k,MeanOmegaCOI4,'cs')
%   hold on

        if(isempty(strfind(nom, 'long'))==0)
            status([4608 :end],4)=0;
        else
            status([4224 :end],4)=0;
        end

end

save(nom,'LabelsGen1','LabelsGen2','LabelsGen3','LabelsGen4',...
    'VectElectGen1','VectElectGen2','VectElectGen3','VectElectGen4',...
    'VectMec','t','status');

clearvars -except FichierSimul k nom NombreDeSimul StabAng StabOme Seuil
disp(k)

end

% Affichage des barres horizontales pour l'angle et la vitesse
% figure(1)
% plot([1;175],[3.14;3.14],[1;175],[-3.14;-3.14])
% figure(2)
% plot([1;175],[Seuil;Seuil],[1;175],[-Seuil;-Seuil])

%% Ajout des différents vecteur de d'entrées

FichierSimul = dir('*.mat');
NombreDeSimul = length(FichierSimul);

% Ajout de cosphi et Phi
for k = 1:NombreDeSimul

    nom=FichierSimul(k,1).name;
    load(nom);
    % Concaténation du vecteur électrique avec l'angle formé entre P et Q

```

```

% des différentes phases pour chaque machines
VectElectGen1=[VectElectGen1 angle(VectElectGen1(:, [3,11,19])+...
j*VectElectGen1(:, [5,13,21]))];
VectElectGen2=[VectElectGen2 angle(VectElectGen2(:, [3,11,19])+...
j*VectElectGen2(:, [5,13,21]))];
VectElectGen3=[VectElectGen3 angle(VectElectGen3(:, [3,11,19])+...
j*VectElectGen3(:, [5,13,21]))];
VectElectGen4=[VectElectGen4 angle(VectElectGen4(:, [3,11,19])+...
j*VectElectGen4(:, [5,13,21]))];

% Concaténation du vecteur électrique avec le cosinus des variables
% calculé précédemment
VectElectGen1=[VectElectGen1 cos(VectElectGen1(:, [40:42]))];
VectElectGen2=[VectElectGen2 cos(VectElectGen2(:, [40:42]))];
VectElectGen3=[VectElectGen3 cos(VectElectGen3(:, [40:42]))];
VectElectGen4=[VectElectGen4 cos(VectElectGen4(:, [40:42]))];

% Ajout des différentes labels pour les 4 générateurs
LabelsGen1=[LabelsGen1 'Gen1/Phase_1/Phi'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_2/Phi'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_3/Phi'];

LabelsGen2=[LabelsGen2 'Gen2/Phase_1/Phi'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_2/Phi'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_3/Phi'];

LabelsGen3=[LabelsGen3 'Gen3/Phase_1/Phi'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_2/Phi'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_3/Phi'];

LabelsGen4=[LabelsGen4 'Gen4/Phase_1/Phi'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_2/Phi'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_3/Phi'];

LabelsGen1=[LabelsGen1 'Gen1/Cos_Phi_1'];
LabelsGen1=[LabelsGen1 'Gen1/Cos_Phi_2'];
LabelsGen1=[LabelsGen1 'Gen1/Cos_Phi_3'];

LabelsGen2=[LabelsGen2 'Gen2/Cos_Phi_1'];
LabelsGen2=[LabelsGen2 'Gen2/Cos_Phi_2'];
LabelsGen2=[LabelsGen2 'Gen2/Cos_Phi_3'];

LabelsGen3=[LabelsGen3 'Gen3/Cos_Phi_1'];
LabelsGen3=[LabelsGen3 'Gen3/Cos_Phi_2'];
LabelsGen3=[LabelsGen3 'Gen3/Cos_Phi_3'];

LabelsGen4=[LabelsGen4 'Gen4/Cos_Phi_1'];
LabelsGen4=[LabelsGen4 'Gen4/Cos_Phi_2'];
LabelsGen4=[LabelsGen4 'Gen4/Cos_Phi_3'];

save(nom, 'LabelsGen1', 'LabelsGen2', 'LabelsGen3', 'LabelsGen4', ...
'VectElectGen1', 'VectElectGen2', 'VectElectGen3', 'VectElectGen4', ...
'VectMec', 't');
end

for k = 1:NombreDeSimul
nom=FichierSimul(k,1).name;
load(nom);

IndiceVectPhi=0;
IndiceVectFreq=0;

```

```

% Détermination, en nombre de cycles des fenetres glissantes pour le
% filtrages
% NOTE, 64 correspond au nombres de cycles, à la fréquence
% d'échantillonnage choisie (3840/60=64cycles)
windowSize1=(1.5*64);
windowSize2=(4*64);
windowSize3=(12*64);
windowSize4=(3*64);
windowSize5=(6*64);
% ----- PMU 1
[~, largGen1]=size(VectElectGen1);
for m = 1:largGen1
    % Détermine ou se situent les tensions et le cosinus phi
    if isempty(strfind(LabelsGen1{1,m}, 'Phase_1/V_RMS'))==0)
        IndicesV1=m;
    end
    if isempty(strfind(LabelsGen1{1,m}, 'Phase_2/V_RMS'))==0)
        IndicesV2=m;
    end
    if isempty(strfind(LabelsGen1{1,m}, 'Phase_3/V_RMS'))==0)
        IndicesV3=m;
    end
    if isempty(strfind(LabelsGen1{1,m}, 'Cos_Phi_1'))==0)
        IndicesCosPhi1=m;
    end
    if isempty(strfind(LabelsGen1{1,m}, 'Cos_Phi_2'))==0)
        IndicesCosPhi2=m;
    end
    if isempty(strfind(LabelsGen1{1,m}, 'Cos_Phi_3'))==0)
        IndicesCosPhi3=m;
    end

    if isempty(strfind(LabelsGen1{1,m}, '/Freq'))==0)
        IndiceVectFreq=[IndiceVectFreq m];
    end
end

IndiceVectPhi(1)=[];
IndiceVectFreq(1)=[];

% Concaténation du vecteur électrique et du produit de la valeur
% absolu de la tension et du cos(phi)
VectElectGen1=[VectElectGen1 abs(VectElectGen1(:,IndicesV1)).*...
    VectElectGen1(:,IndicesCosPhi1)];
VectElectGen1=[VectElectGen1 abs(VectElectGen1(:,IndicesV2)).*...
    VectElectGen1(:,IndicesCosPhi2)];
VectElectGen1=[VectElectGen1 abs(VectElectGen1(:,IndicesV3)).*...
    VectElectGen1(:,IndicesCosPhi3)];
LabelsGen1=[LabelsGen1 'Gen1/Phase_1/UCosPhi'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_2/UCosPhi'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_3/UCosPhi'];

% Concaténation du vecteur électrique avec une dérivée filtrée à 1.5
% cycles du UCosphi
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize1)/...
    windowSize1,1,gradient(VectElectGen1(:,IndicesV1)).*...
    gradient(t([1:length(VectElectGen1)]))));

VectElectGen1=[VectElectGen1 filter(ones(1,windowSize1)/...
    windowSize1,1,gradient(VectElectGen1(:,IndicesV2)).*...
    gradient(t([1:length(VectElectGen1)]))));
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize1)/...
    windowSize1,1,gradient(VectElectGen1(:,IndicesV3)).*...
    gradient(t([1:length(VectElectGen1)]))));
LabelsGen1=[LabelsGen1 'Gen1/Phase_1/dUdtUCosPhi/1.5Cycles'];

```

```

LabelsGen1=[LabelsGen1 'Gen1/Phase_2/dUdtUCosPhi/1.5Cycles'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_3/dUdtUCosPhi/1.5Cycles'];

% Concaténation du vecteur électrique avec une dérivée filtrée à 4
% cycles du UCosphi
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize2)/windowSize2...
    ,1,gradient(VectElectGen1(:,IndicesV1))./...
    gradient(t([1:length(VectElectGen1)]))))];
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize2)/windowSize2...
    ,1,gradient(VectElectGen1(:,IndicesV2))./...
    gradient(t([1:length(VectElectGen1)]))))];
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize2)/windowSize2...
    ,1,gradient(VectElectGen1(:,IndicesV3))./...
    gradient(t([1:length(VectElectGen1)]))))];
LabelsGen1=[LabelsGen1 'Gen1/Phase_1/dUdtUCosPhi/4Cycles'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_2/dUdtUCosPhi/4Cycles'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_3/dUdtUCosPhi/4Cycles'];

% Concaténation du vecteur électrique avec une dérivée filtrée à 12
% cycles du UCosphi
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize3)/windowSize3...
    ,1,gradient(VectElectGen1(:,IndicesV1))./...
    gradient(t([1:length(VectElectGen1)]))))];
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize3)/windowSize3...
    ,1,gradient(VectElectGen1(:,IndicesV2))./...
    gradient(t([1:length(VectElectGen1)]))))];
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize3)/windowSize3...
    ,1,gradient(VectElectGen1(:,IndicesV3))./...
    gradient(t([1:length(VectElectGen1)]))))];
LabelsGen1=[LabelsGen1 'Gen1/Phase_1/dUdtUCosPhi/12Cycles'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_2/dUdtUCosPhi/12Cycles'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_3/dUdtUCosPhi/12Cycles'];

% Concaténation du vecteur électrique avec la différence d'angle
% entre generateur 1 et 2
VectElectGen1=[VectElectGen1 VectElectGen1(:,IndiceVectFreq(1))...
    -VectElectGen2(:,IndiceVectFreq(1))];
VectElectGen1=[VectElectGen1 VectElectGen1(:,IndiceVectFreq(2))...
    -VectElectGen2(:,IndiceVectFreq(2))];
VectElectGen1=[VectElectGen1 VectElectGen1(:,IndiceVectFreq(3))...
    -VectElectGen2(:,IndiceVectFreq(3))];
LabelsGen1=[LabelsGen1 'Gen1/Phase_1/Freq_Gen1_Gen2'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_2/Freq_Gen1_Gen2'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_3/Freq_Gen1_Gen2'];

% Concaténation du vecteur électrique avec la différence d'angle
% entre generateur 1 et 3
VectElectGen1=[VectElectGen1 VectElectGen1(:,IndiceVectFreq(1))...
    -VectElectGen3(:,IndiceVectFreq(1))];
VectElectGen1=[VectElectGen1 VectElectGen1(:,IndiceVectFreq(2))...
    -VectElectGen3(:,IndiceVectFreq(2))];
VectElectGen1=[VectElectGen1 VectElectGen1(:,IndiceVectFreq(3))...
    -VectElectGen3(:,IndiceVectFreq(3))];
LabelsGen1=[LabelsGen1 'Gen1/Phase_1/Freq_Gen1_Gen3'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_2/Freq_Gen1_Gen3'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_3/Freq_Gen1_Gen3'];

% Concaténation du vecteur électrique avec la différence d'angle
% entre generateur 1 et 4
VectElectGen1=[VectElectGen1 VectElectGen1(:,IndiceVectFreq(1))...
    -VectElectGen4(:,IndiceVectFreq(1))];
VectElectGen1=[VectElectGen1 VectElectGen1(:,IndiceVectFreq(2))...
    -VectElectGen4(:,IndiceVectFreq(2))];
VectElectGen1=[VectElectGen1 VectElectGen1(:,IndiceVectFreq(3))...

```

```

-VectElectGen4(:,IndiceVectFreq(3))];
LabelsGen1=[LabelsGen1 'Gen1/Phase_1/Freq_Gen1_Gen4'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_2/Freq_Gen1_Gen4'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_3/Freq_Gen1_Gen4'];

% Concaténation du vecteur électrique la fréquence à la barre filtrée à
% 3 cycles
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize4)/windowSize4...
,1,VectElectGen1(:,IndiceVectFreq(1))))];
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize4)/windowSize4...
,1,VectElectGen1(:,IndiceVectFreq(2))))];
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize4)/windowSize4...
,1,VectElectGen1(:,IndiceVectFreq(3))))];
LabelsGen1=[LabelsGen1 'Gen1/Phase_1/Freq_Filt_3cy'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_2/Freq_Filt_3cy'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_3/Freq_Filt_3cy'];

% Concaténation du vecteur électrique la fréquence à la barre filtrée à
% 6 cycles
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize5)/windowSize5...
,1,VectElectGen1(:,IndiceVectFreq(1))))];
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize5)/windowSize5...
,1,VectElectGen1(:,IndiceVectFreq(2))))];
VectElectGen1=[VectElectGen1 filter(ones(1,windowSize5)/windowSize5...
,1,VectElectGen1(:,IndiceVectFreq(3))))];
LabelsGen1=[LabelsGen1 'Gen1/Phase_1/Freq_Filt_6cy'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_2/Freq_Filt_6cy'];
LabelsGen1=[LabelsGen1 'Gen1/Phase_3/Freq_Filt_6cy'];

clear Indices*

% ----- PMU 2
[~, largGen2]=size(VectElectGen2);
for m = 1:largGen2
    if(isempty(strfind(LabelsGen2{1,m}, 'Phase_1/V_RMS'))==0)
        IndicesV1=m;
    end
    if(isempty(strfind(LabelsGen2{1,m}, 'Phase_2/V_RMS'))==0)
        IndicesV2=m;
    end
    if(isempty(strfind(LabelsGen2{1,m}, 'Phase_3/V_RMS'))==0)
        IndicesV3=m;
    end
    if(isempty(strfind(LabelsGen2{1,m}, 'Cos_Phi_1'))==0)
        IndicesCosPhi1=m;
    end
    if(isempty(strfind(LabelsGen2{1,m}, 'Cos_Phi_2'))==0)
        IndicesCosPhi2=m;
    end
    if(isempty(strfind(LabelsGen2{1,m}, 'Cos_Phi_3'))==0)
        IndicesCosPhi3=m;
    end
end

VectElectGen2=[VectElectGen2 abs(VectElectGen2(:,IndicesV1)).*...
    VectElectGen2(:,IndicesCosPhi1)];
VectElectGen2=[VectElectGen2 abs(VectElectGen2(:,IndicesV2)).*...
    VectElectGen2(:,IndicesCosPhi2)];

```

```

VectElectGen2=[VectElectGen2 abs(VectElectGen2(:,IndicesV3)).*...
    VectElectGen2(:,IndicesCosPhi3)];
LabelsGen2=[LabelsGen2 'Gen2/Phase_1/UCosPhi'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_2/UCosPhi'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_3/UCosPhi'];

% Derivee de Ucosphi

VectElectGen2=[VectElectGen2 filter(ones(1,windowSize1)/windowSize1...
    ,1,gradient(VectElectGen2(:,IndicesV1))./...
    gradient(t([1:length(VectElectGen2)]))))];
VectElectGen2=[VectElectGen2 filter(ones(1,windowSize1)/windowSize1...
    ,1,gradient(VectElectGen2(:,IndicesV2))./...
    gradient(t([1:length(VectElectGen2)]))))];
VectElectGen2=[VectElectGen2 filter(ones(1,windowSize1)/windowSize1...
    ,1,gradient(VectElectGen2(:,IndicesV3))./...
    gradient(t([1:length(VectElectGen2)]))))];
LabelsGen2=[LabelsGen2 'Gen2/Phase_1/dUdtUCosPhi/1.5Cycles'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_2/dUdtUCosPhi/1.5Cycles'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_3/dUdtUCosPhi/1.5Cycles'];

VectElectGen2=[VectElectGen2 filter(ones(1,windowSize2)/windowSize2...
    ,1,gradient(VectElectGen2(:,IndicesV1))./...
    gradient(t([1:length(VectElectGen2)]))))];
VectElectGen2=[VectElectGen2 filter(ones(1,windowSize2)/windowSize2...
    ,1,gradient(VectElectGen2(:,IndicesV2))./...
    gradient(t([1:length(VectElectGen2)]))))];
VectElectGen2=[VectElectGen2 filter(ones(1,windowSize2)/windowSize2...
    ,1,gradient(VectElectGen2(:,IndicesV3))./...
    gradient(t([1:length(VectElectGen2)]))))];
LabelsGen2=[LabelsGen2 'Gen2/Phase_1/dUdtUCosPhi/4Cycles'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_2/dUdtUCosPhi/4Cycles'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_3/dUdtUCosPhi/4Cycles'];

VectElectGen2=[VectElectGen2 filter(ones(1,windowSize3)/windowSize3...
    ,1,gradient(VectElectGen2(:,IndicesV1))./...
    gradient(t([1:length(VectElectGen2)]))))];
VectElectGen2=[VectElectGen2 filter(ones(1,windowSize3)/windowSize3...
    ,1,gradient(VectElectGen2(:,IndicesV2))./...
    gradient(t([1:length(VectElectGen2)]))))];
VectElectGen2=[VectElectGen2 filter(ones(1,windowSize3)/windowSize3...
    ,1,gradient(VectElectGen2(:,IndicesV3))./...
    gradient(t([1:length(VectElectGen2)]))))];
LabelsGen2=[LabelsGen2 'Gen2/Phase_1/dUdtUCosPhi/12Cycles'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_2/dUdtUCosPhi/12Cycles'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_3/dUdtUCosPhi/12Cycles'];

% différence de frequence entre generateur 2 et 1
VectElectGen2=[VectElectGen2 VectElectGen2(:,IndiceVectFreq(1))...
    -VectElectGen1(:,IndiceVectFreq(1))];
VectElectGen2=[VectElectGen2 VectElectGen2(:,IndiceVectFreq(2))...
    -VectElectGen1(:,IndiceVectFreq(2))];
VectElectGen2=[VectElectGen2 VectElectGen2(:,IndiceVectFreq(3))...
    -VectElectGen1(:,IndiceVectFreq(3))];
LabelsGen2=[LabelsGen2 'Gen2/Phase_1/Freq_Gen2_Gen1'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_2/Freq_Gen2_Gen1'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_3/Freq_Gen2_Gen1'];
% différence de frequence entre generateur 2 et 3
VectElectGen2=[VectElectGen2 VectElectGen2(:,IndiceVectFreq(1))...
    -VectElectGen3(:,IndiceVectFreq(1))];
VectElectGen2=[VectElectGen2 VectElectGen2(:,IndiceVectFreq(2))...
    -VectElectGen3(:,IndiceVectFreq(2))];
VectElectGen2=[VectElectGen2 VectElectGen2(:,IndiceVectFreq(3))...
    -VectElectGen3(:,IndiceVectFreq(3))];

```

```

-VectElectGen3(:,IndiceVectFreq(3))];
LabelsGen2=[LabelsGen2 'Gen2/Phase_1/Freq_Gen2_Gen3'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_2/Freq_Gen2_Gen3'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_3/Freq_Gen2_Gen3'];
% différence de frequence entre generateur 2 et 4
VectElectGen2=[VectElectGen2 VectElectGen2(:,IndiceVectFreq(1))...
-VectElectGen4(:,IndiceVectFreq(1))];
VectElectGen2=[VectElectGen2 VectElectGen2(:,IndiceVectFreq(2))...
-VectElectGen4(:,IndiceVectFreq(2))];
VectElectGen2=[VectElectGen2 VectElectGen2(:,IndiceVectFreq(3))...
-VectElectGen4(:,IndiceVectFreq(3))];
LabelsGen2=[LabelsGen2 'Gen2/Phase_1/Freq_Gen2_Gen4'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_2/Freq_Gen2_Gen4'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_3/Freq_Gen2_Gen4'];

VectElectGen2=[VectElectGen2 filter(ones(1,windowSize4)/windowSize4...
,1,VectElectGen2(:,IndiceVectFreq(1)))];
VectElectGen2=[VectElectGen2 filter(ones(1,windowSize4)/windowSize4...
,1,VectElectGen2(:,IndiceVectFreq(2)))];
VectElectGen2=[VectElectGen2 filter(ones(1,windowSize4)/windowSize4...
,1,VectElectGen2(:,IndiceVectFreq(3)))];
LabelsGen2=[LabelsGen2 'Gen2/Phase_1/Freq_Filt_3cy'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_2/Freq_Filt_3cy'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_3/Freq_Filt_3cy'];

VectElectGen2=[VectElectGen2 filter(ones(1,windowSize5)/windowSize5...
,1,VectElectGen2(:,IndiceVectFreq(1)))];
VectElectGen2=[VectElectGen2 filter(ones(1,windowSize5)/windowSize5...
,1,VectElectGen2(:,IndiceVectFreq(2)))];
VectElectGen2=[VectElectGen2 filter(ones(1,windowSize5)/windowSize5...
,1,VectElectGen2(:,IndiceVectFreq(3)))];
LabelsGen2=[LabelsGen2 'Gen2/Phase_1/Freq_Filt_6cy'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_2/Freq_Filt_6cy'];
LabelsGen2=[LabelsGen2 'Gen2/Phase_3/Freq_Filt_6cy'];

clear Indices*

% ----- PMU 3
[~, largGen3]=size(VectElectGen3);
for m = 1:largGen3
    if(isempty(strfind(LabelsGen3{1,m}, 'Phase_1/V_RMS'))==0)
        IndicesV1=m;
    end
    if(isempty(strfind(LabelsGen3{1,m}, 'Phase_2/V_RMS'))==0)
        IndicesV2=m;
    end
    if(isempty(strfind(LabelsGen3{1,m}, 'Phase_3/V_RMS'))==0)
        IndicesV3=m;
    end
    if(isempty(strfind(LabelsGen3{1,m}, 'Cos_Phi_1'))==0)
        IndicesCosPhi1=m;
    end
    if(isempty(strfind(LabelsGen3{1,m}, 'Cos_Phi_2'))==0)
        IndicesCosPhi2=m;
    end
    if(isempty(strfind(LabelsGen3{1,m}, 'Cos_Phi_3'))==0)
        IndicesCosPhi3=m;
    end
end

end

```

```

VectElectGen3=[VectElectGen3 abs(VectElectGen3(:,IndicesV1)).*...
    VectElectGen3(:,IndicesCosPhi1)];
VectElectGen3=[VectElectGen3 abs(VectElectGen3(:,IndicesV2)).*...
    VectElectGen3(:,IndicesCosPhi2)];
VectElectGen3=[VectElectGen3 abs(VectElectGen3(:,IndicesV3)).*...
    VectElectGen3(:,IndicesCosPhi3)];
LabelsGen3=[LabelsGen3 'Gen3/Phase_1/UCosPhi'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_2/UCosPhi'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_3/UCosPhi'];

% Derivee de Ucosphi

VectElectGen3=[VectElectGen3 filter(ones(1,windowSize1)/windowSize1...
    ,1,gradient(VectElectGen3(:,IndicesV1))./...
    gradient(t([1:length(VectElectGen3)])))]];
VectElectGen3=[VectElectGen3 filter(ones(1,windowSize1)/windowSize1...
    ,1,gradient(VectElectGen3(:,IndicesV2))./...
    gradient(t([1:length(VectElectGen3)])))]];
VectElectGen3=[VectElectGen3 filter(ones(1,windowSize1)/windowSize1...
    ,1,gradient(VectElectGen3(:,IndicesV3))./...
    gradient(t([1:length(VectElectGen3)])))]];
LabelsGen3=[LabelsGen3 'Gen3/Phase_1/dUdtUCosPhi/1.5Cycles'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_2/dUdtUCosPhi/1.5Cycles'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_3/dUdtUCosPhi/1.5Cycles'];

VectElectGen3=[VectElectGen3 filter(ones(1,windowSize2)/windowSize2...
    ,1,gradient(VectElectGen3(:,IndicesV1))./...
    gradient(t([1:length(VectElectGen3)])))]];
VectElectGen3=[VectElectGen3 filter(ones(1,windowSize2)/windowSize2...
    ,1,gradient(VectElectGen3(:,IndicesV2))./...
    gradient(t([1:length(VectElectGen3)])))]];
VectElectGen3=[VectElectGen3 filter(ones(1,windowSize2)/windowSize2...
    ,1,gradient(VectElectGen3(:,IndicesV3))./...
    gradient(t([1:length(VectElectGen3)])))]];
LabelsGen3=[LabelsGen3 'Gen3/Phase_1/dUdtUCosPhi/4Cycles'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_2/dUdtUCosPhi/4Cycles'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_3/dUdtUCosPhi/4Cycles'];

VectElectGen3=[VectElectGen3 filter(ones(1,windowSize3)/windowSize3...
    ,1,gradient(VectElectGen3(:,IndicesV1))./...
    gradient(t([1:length(VectElectGen3)])))]];
VectElectGen3=[VectElectGen3 filter(ones(1,windowSize3)/windowSize3...
    ,1,gradient(VectElectGen3(:,IndicesV2))./...
    gradient(t([1:length(VectElectGen3)])))]];
VectElectGen3=[VectElectGen3 filter(ones(1,windowSize3)/windowSize3...
    ,1,gradient(VectElectGen3(:,IndicesV3))./...
    gradient(t([1:length(VectElectGen3)])))]];
LabelsGen3=[LabelsGen3 'Gen3/Phase_1/dUdtUCosPhi/12Cycles'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_2/dUdtUCosPhi/12Cycles'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_3/dUdtUCosPhi/12Cycles'];

% différence de frequence entre generateur 3 et 1
VectElectGen3=[VectElectGen3 VectElectGen3(:,IndiceVectFreq(1))-...
    VectElectGen1(:,IndiceVectFreq(1))];
VectElectGen3=[VectElectGen3 VectElectGen3(:,IndiceVectFreq(2))-...
    VectElectGen1(:,IndiceVectFreq(2))];
VectElectGen3=[VectElectGen3 VectElectGen3(:,IndiceVectFreq(3))-...
    VectElectGen1(:,IndiceVectFreq(3))];
LabelsGen3=[LabelsGen3 'Gen3/Phase_1/Freq_Gen3_Gen1'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_2/Freq_Gen3_Gen1'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_3/Freq_Gen3_Gen1'];
% différence de frequence entre generateur 3 et 2

```

```

VectElectGen3=[VectElectGen3 VectElectGen3(:,IndiceVectFreq(1))-...
    VectElectGen2(:,IndiceVectFreq(1))];
VectElectGen3=[VectElectGen3 VectElectGen3(:,IndiceVectFreq(2))-...
    VectElectGen2(:,IndiceVectFreq(2))];
VectElectGen3=[VectElectGen3 VectElectGen3(:,IndiceVectFreq(3))-...
    VectElectGen2(:,IndiceVectFreq(3))];
LabelsGen3=[LabelsGen3 'Gen3/Phase_1/Freq_Gen3_Gen2'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_2/Freq_Gen3_Gen2'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_3/Freq_Gen3_Gen2'];
% différence de frequence entre generateur 3 et 4
VectElectGen3=[VectElectGen3 VectElectGen3(:,IndiceVectFreq(1))-...
    VectElectGen4(:,IndiceVectFreq(1))];
VectElectGen3=[VectElectGen3 VectElectGen3(:,IndiceVectFreq(2))-...
    VectElectGen4(:,IndiceVectFreq(2))];
VectElectGen3=[VectElectGen3 VectElectGen3(:,IndiceVectFreq(3))-...
    VectElectGen4(:,IndiceVectFreq(3))];
LabelsGen3=[LabelsGen3 'Gen3/Phase_1/Freq_Gen3_Gen4'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_2/Freq_Gen3_Gen4'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_3/Freq_Gen3_Gen4'];

VectElectGen3=[VectElectGen3 filter(ones(1,windowSize4)/windowSize4...
    ,1,VectElectGen3(:,IndiceVectFreq(1)))];
VectElectGen3=[VectElectGen3 filter(ones(1,windowSize4)/windowSize4...
    ,1,VectElectGen3(:,IndiceVectFreq(2)))];
VectElectGen3=[VectElectGen3 filter(ones(1,windowSize4)/windowSize4...
    ,1,VectElectGen3(:,IndiceVectFreq(3)))];
LabelsGen3=[LabelsGen3 'Gen3/Phase_1/Freq_Filt_3cy'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_2/Freq_Filt_3cy'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_3/Freq_Filt_3cy'];

VectElectGen3=[VectElectGen3 filter(ones(1,windowSize5)/windowSize5...
    ,1,VectElectGen3(:,IndiceVectFreq(1)))];
VectElectGen3=[VectElectGen3 filter(ones(1,windowSize5)/windowSize5...
    ,1,VectElectGen3(:,IndiceVectFreq(2)))];
VectElectGen3=[VectElectGen3 filter(ones(1,windowSize5)/windowSize5...
    ,1,VectElectGen3(:,IndiceVectFreq(3)))];
LabelsGen3=[LabelsGen3 'Gen3/Phase_1/Freq_Filt_6cy'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_2/Freq_Filt_6cy'];
LabelsGen3=[LabelsGen3 'Gen3/Phase_3/Freq_Filt_6cy'];

clear Indices*

% ----- PMU 4
[~, largGen4]=size(VectElectGen4);
for m = 1:largGen4
    if isempty(strfind(LabelsGen4{1,m}, 'Phase_1/V_RMS')==0)
        IndicesV1=m;
    end
    if isempty(strfind(LabelsGen4{1,m}, 'Phase_2/V_RMS')==0)
        IndicesV2=m;
    end
    if isempty(strfind(LabelsGen4{1,m}, 'Phase_3/V_RMS')==0)
        IndicesV3=m;
    end
    if isempty(strfind(LabelsGen4{1,m}, 'Cos_Phi_1')==0)
        IndicesCosPhi1=m;
    end
    if isempty(strfind(LabelsGen4{1,m}, 'Cos_Phi_2')==0)
        IndicesCosPhi2=m;
    end
    if isempty(strfind(LabelsGen4{1,m}, 'Cos_Phi_3')==0)
        IndicesCosPhi3=m;
    end
end

```

```

end

end

VectElectGen4=[VectElectGen4 abs(VectElectGen4(:,IndicesV1)).*...
    VectElectGen4(:,IndicesCosPhi1)];
VectElectGen4=[VectElectGen4 abs(VectElectGen4(:,IndicesV2)).*...
    VectElectGen4(:,IndicesCosPhi2)];
VectElectGen4=[VectElectGen4 abs(VectElectGen4(:,IndicesV3)).*...
    VectElectGen4(:,IndicesCosPhi3)];
LabelsGen4=[LabelsGen4 'Gen4/Phase_1/UCosPhi'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_2/UCosPhi'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_3/UCosPhi'];

% Derivee de Ucosphi

VectElectGen4=[VectElectGen4 filter(ones(1,windowSize1)/windowSize1...
    ,1,gradient(VectElectGen4(:,IndicesV1))./...
    gradient(t([1:length(VectElectGen4)])))]];
VectElectGen4=[VectElectGen4 filter(ones(1,windowSize1)/windowSize1...
    ,1,gradient(VectElectGen4(:,IndicesV2))./...
    gradient(t([1:length(VectElectGen4)])))]];
VectElectGen4=[VectElectGen4 filter(ones(1,windowSize1)/windowSize1...
    ,1,gradient(VectElectGen4(:,IndicesV3))./...
    gradient(t([1:length(VectElectGen4)])))]];
LabelsGen4=[LabelsGen4 'Gen4/Phase_1/dUdtUCosPhi/1.5Cycles'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_2/dUdtUCosPhi/1.5Cycles'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_3/dUdtUCosPhi/1.5Cycles'];

VectElectGen4=[VectElectGen4 filter(ones(1,windowSize2)/windowSize2...
    ,1,gradient(VectElectGen4(:,IndicesV1))./...
    gradient(t([1:length(VectElectGen4)])))]];
VectElectGen4=[VectElectGen4 filter(ones(1,windowSize2)/windowSize2...
    ,1,gradient(VectElectGen4(:,IndicesV2))./...
    gradient(t([1:length(VectElectGen4)])))]];
VectElectGen4=[VectElectGen4 filter(ones(1,windowSize2)/windowSize2...
    ,1,gradient(VectElectGen4(:,IndicesV3))./...
    gradient(t([1:length(VectElectGen4)])))]];
LabelsGen4=[LabelsGen4 'Gen4/Phase_1/dUdtUCosPhi/4Cycles'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_2/dUdtUCosPhi/4Cycles'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_3/dUdtUCosPhi/4Cycles'];

VectElectGen4=[VectElectGen4 filter(ones(1,windowSize3)/windowSize3...
    ,1,gradient(VectElectGen4(:,IndicesV1))./...
    gradient(t([1:length(VectElectGen4)])))]];
VectElectGen4=[VectElectGen4 filter(ones(1,windowSize3)/windowSize3...
    ,1,gradient(VectElectGen4(:,IndicesV2))./...
    gradient(t([1:length(VectElectGen4)])))]];
VectElectGen4=[VectElectGen4 filter(ones(1,windowSize3)/windowSize3...
    ,1,gradient(VectElectGen4(:,IndicesV3))./...
    gradient(t([1:length(VectElectGen4)])))]];
LabelsGen4=[LabelsGen4 'Gen4/Phase_1/dUdtUCosPhi/12Cycles'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_2/dUdtUCosPhi/12Cycles'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_3/dUdtUCosPhi/12Cycles'];

% difference de frequence entre generateur 4 et 1
VectElectGen4=[VectElectGen4 VectElectGen4(:,IndiceVectFreq(1))...
    -VectElectGen1(:,IndiceVectFreq(1))];
VectElectGen4=[VectElectGen4 VectElectGen4(:,IndiceVectFreq(2))...
    -VectElectGen1(:,IndiceVectFreq(2))];
VectElectGen4=[VectElectGen4 VectElectGen4(:,IndiceVectFreq(3))...

```

```

-VectElectGen1(:,IndiceVectFreq(3)));
LabelsGen4=[LabelsGen4 'Gen4/Phase_1/Freq_Gen4_Gen1'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_2/Freq_Gen4_Gen1'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_3/Freq_Gen4_Gen1'];
% différence de frequence entre generateur 4 et 2
VectElectGen4=[VectElectGen4 VectElectGen4(:,IndiceVectFreq(1))...
-VectElectGen2(:,IndiceVectFreq(1))];
VectElectGen4=[VectElectGen4 VectElectGen4(:,IndiceVectFreq(2))...
-VectElectGen2(:,IndiceVectFreq(2))];
VectElectGen4=[VectElectGen4 VectElectGen4(:,IndiceVectFreq(3))...
-VectElectGen2(:,IndiceVectFreq(3))];
LabelsGen4=[LabelsGen4 'Gen4/Phase_1/Freq_Gen4_Gen2'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_2/Freq_Gen4_Gen2'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_3/Freq_Gen4_Gen2'];
% différence de frequence entre generateur 4 et 3
VectElectGen4=[VectElectGen4 VectElectGen4(:,IndiceVectFreq(1))...
-VectElectGen3(:,IndiceVectFreq(1))];
VectElectGen4=[VectElectGen4 VectElectGen4(:,IndiceVectFreq(2))...
-VectElectGen3(:,IndiceVectFreq(2))];
VectElectGen4=[VectElectGen4 VectElectGen4(:,IndiceVectFreq(3))...
-VectElectGen3(:,IndiceVectFreq(3))];
LabelsGen4=[LabelsGen4 'Gen4/Phase_1/Freq_Gen4_Gen3'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_2/Freq_Gen4_Gen3'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_3/Freq_Gen4_Gen3'];

VectElectGen4=[VectElectGen4 filter(ones(1,windowSize4)/windowSize4...
,1,VectElectGen4(:,IndiceVectFreq(1)))];
VectElectGen4=[VectElectGen4 filter(ones(1,windowSize4)/windowSize4...
,1,VectElectGen4(:,IndiceVectFreq(2)))];
VectElectGen4=[VectElectGen4 filter(ones(1,windowSize4)/windowSize4...
,1,VectElectGen4(:,IndiceVectFreq(3)))];
LabelsGen4=[LabelsGen4 'Gen4/Phase_1/Freq_Filt_3cy'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_2/Freq_Filt_3cy'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_3/Freq_Filt_3cy'];

VectElectGen4=[VectElectGen4 filter(ones(1,windowSize5)/windowSize5...
,1,VectElectGen4(:,IndiceVectFreq(1)))];
VectElectGen4=[VectElectGen4 filter(ones(1,windowSize5)/windowSize5...
,1,VectElectGen4(:,IndiceVectFreq(2)))];
VectElectGen4=[VectElectGen4 filter(ones(1,windowSize5)/windowSize5...
,1,VectElectGen4(:,IndiceVectFreq(3)))];
LabelsGen4=[LabelsGen4 'Gen4/Phase_1/Freq_Filt_6cy'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_2/Freq_Filt_6cy'];
LabelsGen4=[LabelsGen4 'Gen4/Phase_3/Freq_Filt_6cy'];

clear Indices*

save(nom,'LabelsGen1','LabelsGen2','LabelsGen3','LabelsGen4',...
'VectElectGen1','VectElectGen2','VectElectGen3','VectElectGen4',...
'VectMec','t');

end

```

Annexe 6

Code MATLAB pour l'exportation des données dans un format CSV

```

%% Histogramme des cas stables et instables

% Repertoire les fichiers .mat
FichierSimul = dir('*.mat');
% Détermine le nombre de simulations
NombreDeSimul = length(FichierSimul);

Sim_Sta_Inst=[0 0 0 0];

% Parcours toutes les simulations et cherche, dans le vecteur staus, les
% simulations stables (0) ou les simulations instables (1)
for k = 1:NombreDeSimul
    nom=FichierSimul(k,1).name;
    load(nom,'status');

    if(find(status(:,1)==1)) % Si la colonne contient un 1, Instable
        Sim_Sta_Inst(k,1)=1;
        sim{k,1}=nom;
    else
        Sim_Sta_Inst(k,1)=0; % Si la colonne contient un 0, Stable
    end

    if(find(status(:,2)==1))
        Sim_Sta_Inst(k,2)=1;
        sim{k,2}=nom;
    else
        Sim_Sta_Inst(k,2)=0;
    end

    if(find(status(:,3)==1))
        Sim_Sta_Inst(k,3)=1;
        sim{k,3}=nom;
    else
        Sim_Sta_Inst(k,3)=0;
    end

    if(find(status(:,4)==1))
        Sim_Sta_Inst(k,4)=1;
        sim{k,4}=nom;
    else
        Sim_Sta_Inst(k,4)=0;
    end
end

Prop=histc(Sim_Sta_Inst,0:1:1); %Proportion des fichier stables et instables

%% Creation des fichiers de validation de test et d'apprenissage Gen 1

```

```

% Ce script est utilisé pour concatener les simulations d'apprentissage de
% test et de validation pour le générateur 1

% Pour Gen 1

RandIDGen1=1234; % Fixe les constantes random pour Gen 1, 2, 3 et 4
RandIDGen2=1068;
RandIDGen3=8214;
RandIDGen4=3422;
DownS=2;          % Constante d'échantillonnage
TempsApp=5;        % Longueur de la fenêtre pour l'ensemble d'apprentissage
TempsTest=5;       % pour l'ensemble de Test
TempsVal=5;        % pour l'ensemble de Validation

PropApp=int16(Prop*0.6);          % Ratio du nombre de simulations
                                   % pour l'ensemble d'apprentissage
PropTest=int16(Prop*0.15);       % Ratio pour l'ensemble de test
PropVal=int16(Prop)-PropApp-PropTest; % Le reste des simulations est placé dans
                                   % l'ensemble de validation

FichierSimul = dir('*.mat');
NombreDeSimul = length(FichierSimul);

% Initialisation des vecteurs servant à la concaténation
DataAppGen1_end=zeros(1,72);
DataAppStatusGen1_end=0;
DataTestGen1_end=zeros(1,72);
DataTestStatusGen1_end=0;
DataValGen1_end=zeros(1,72);
DataValStatusGen1_end=0;

rand('twister', RandIDGen1);
% Echantillonnage parmi les différentes simulation
VectAppInstab=sort(randsample(find(Sim_Sta_Inst(:,1)==1),PropApp(2,1)));
rand('twister', RandIDGen1);
VectAppSta=sort(randsample(find(Sim_Sta_Inst(:,1)==0),PropApp(1,1)));
% 999 est inséré lorsque une simulation a déjà été tirée
Sim_Sta_Inst(VectAppSta,1)=999;
Sim_Sta_Inst(VectAppInstab,1)=999;
rand('twister', RandIDGen1);
VectTestInstab=sort(randsample(find(Sim_Sta_Inst(:,1)==1),PropTest(2,1)));
rand('twister', RandIDGen1);
VectTestSta=sort(randsample(find(Sim_Sta_Inst(:,1)==0),PropTest(1,1)));
Sim_Sta_Inst(VectTestSta,1)=999;
Sim_Sta_Inst(VectTestInstab,1)=999;
rand('twister', RandIDGen1);
VectValInstab=sort(randsample(find(Sim_Sta_Inst(:,1)==1),PropVal(2,1)));
rand('twister', RandIDGen1);
VectValSta=sort(randsample(find(Sim_Sta_Inst(:,1)==0),PropVal(1,1)));
Sim_Sta_Inst(VectValSta,1)=999;
Sim_Sta_Inst(VectValInstab,1)=999;

for k = 1:NombreDeSimul
    % Vérifie si pour la simulation numéro "k", celle-ci doit aller dans
    % le vecteur d'apprentissage
    if (isempty(find(VectAppSta(:,1)==k))~=1 || ...
        isempty(find(VectAppInstab(:,1)==k))~=1)

        % Sauvegarde et charge le nom de la simulation ciblée
        nom=FichierSimul(k,1).name;
        Gen1{k,1}=nom;
        Gen1{k,2}=length(DataAppStatusGen1_end);
        load(nom, 'status', 'VectElectGen1');
    end
end

```

```

% vérifie en fonction du nom, si le CC est long ou court
if isempty(strfind(nom, 'long'))==0
    % Vérifie si la simulation est plus longue que la fenêtre
    % d'apprentissage
    if length(VectElectGen1)>(4608)+3840*TempsApp
        % Concaténation de la simulation à partir du court-circuit
        % jusqu'à la fenêtre défini par l'utilisateur
        DataAppGen1_end=[DataAppGen1_end;downsample(VectElectGen1(...
            [4608:(4608)+3840*TempsApp],:),DownS)];
        DataAppStatusGen1_end=[DataAppStatusGen1_end;downsample(...
            status([4608:(4608)+3840*TempsApp],1),DownS)];
    else
        % Concaténation de la simulation si elle est plus courte
        % que la fenêtre
        DataAppGen1_end=[DataAppGen1_end;downsample(...
            VectElectGen1([4608:end],:),DownS)];
        DataAppStatusGen1_end=[DataAppStatusGen1_end;downsample(...
            status([4608:end],1),DownS)];
    end

else
    % Entre si la simulation est courte
    % Vérifie que la simulation longue que la fenêtre
    % d'apprentissage
    if length(VectElectGen1)>(4224)+3840*TempsApp

        DataAppGen1_end=[DataAppGen1_end;downsample(VectElectGen1(...
            [4224:(4224)+3840*TempsApp],:),DownS)];
        DataAppStatusGen1_end=[DataAppStatusGen1_end;downsample(...
            status([4224:(4224)+3840*TempsApp],1),DownS)];

    else
        % entre si elle est courte
        DataAppGen1_end=[DataAppGen1_end;downsample(...
            VectElectGen1([4224:end],:),DownS)];
        DataAppStatusGen1_end=[DataAppStatusGen1_end;downsample(...
            status([4224:end],1),DownS)];
    end
end

% La même chose est effectué pour le vecteur de test
elseif(isempty(find(VectTestSta(:,1)==k))~=1||...
    isempty(find(VectTestInstab(:,1)==k))~=1)
    nom=FichierSimul(k,1).name;
    Gen1{k,3}=nom;
    Gen1{k,4}=length(DataTestStatusGen1_end);
    load(nom,'status','VectElectGen1');
    if(isempty(strfind(nom, 'long'))==0)
        if length(VectElectGen1)>(4608)+3840*TempsTest
            DataTestGen1_end=[DataTestGen1_end;downsample(...
                VectElectGen1([4608:(4608)+3840*TempsTest],:),DownS)];
            DataTestStatusGen1_end=[DataTestStatusGen1_end;downsample(...
                status([4608:(4608)+3840*TempsTest],1),DownS)];
        else
            DataTestGen1_end=[DataTestGen1_end;downsample(...
                VectElectGen1([4608:end],:),DownS)];
            DataTestStatusGen1_end=[DataTestStatusGen1_end;...
                downsample(status([4608:end],1),DownS)];
        end
    end

else
    if length(VectElectGen1)>(4224)+3840*TempsTest

        DataTestGen1_end=[DataTestGen1_end;downsample(...
            VectElectGen1([4224:(4224)+3840*TempsTest],:),DownS)];
        DataTestStatusGen1_end=[DataTestStatusGen1_end;downsample(...
            status([4224:(4224)+3840*TempsTest],1),DownS)];
    else
        DataTestGen1_end=[DataTestGen1_end;downsample(...

```

```

        VectElectGen1([4224:end],:),DownS)];
        DataTestStatusGen1_end=[DataTestStatusGen1_end;...
            downsample(status([4224:end],1),DownS)];
    end
end

elseif(isempty(find(VectValSta(:,1)==k))~=1||...
    isempty(find(VectValInstab(:,1)==k))~=1)

    nom=FichierSimul(k,1).name;
    Gen1{k,5}=nom;
    Gen1{k,6}=length(DataValStatusGen1_end);
    load(nom,'status','VectElectGen1');
    if(isempty(strfind(nom,'long'))==0)
        if length(VectElectGen1)>(4608)+3840*TempsVal
            DataValGen1_end=[DataValGen1_end;downsample(...
                VectElectGen1([4608:(4608)+3840*TempsVal],:),DownS)];
            DataValStatusGen1_end=[DataValStatusGen1_end;downsample(...
                status([4608:(4608)+3840*TempsVal],1),DownS)];
        else
            DataValGen1_end=[DataValGen1_end;downsample(...
                VectElectGen1([4608:end],:),DownS)];
            DataValStatusGen1_end=[DataValStatusGen1_end;...
                downsample(status([4608:end],1),DownS)];
        end
    else
        if length(VectElectGen1)>(4224)+3840*TempsVal

            DataValGen1_end=[DataValGen1_end;downsample(VectElectGen1...
                ([4224:(4224)+3840*TempsVal],:),DownS)];
            DataValStatusGen1_end=[DataValStatusGen1_end;downsample(...
                status([4224:(4224)+3840*TempsVal],1),DownS)];
        else
            DataValGen1_end=[DataValGen1_end;downsample(VectElectGen1...
                ([4224:end],:),DownS)];
            DataValStatusGen1_end=[DataValStatusGen1_end;downsample(...
                status([4224:end],1),DownS)];
        end
    end
end

end

end

DataAppGen1_end=DataAppGen1_end([2:end],:);
DataAppStatusGen1_end=DataAppStatusGen1_end([2:end],:);
load('rpsslackg1_1001_tripahse_ligne30_long.mat','LabelsGen1');
LabelsGen1{1,73}='Status';
csvwrite_with_headers(strcat('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\DataAppGen1_',num2str(TempsApp),'s.csv'),[DataAppGen1_end
DataAppStatusGen1_end],LabelsGen1);
DataTestGen1_end=DataTestGen1_end([2:end],:);
DataTestStatusGen1_end=DataTestStatusGen1_end([2:end],:);
csvwrite_with_headers(strcat('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\DataTestGen1_',num2str(TempsTest),'s.csv'),[DataTestGen1_end
DataTestStatusGen1_end],LabelsGen1);
DataValGen1_end=DataValGen1_end([2:end],:);
DataValStatusGen1_end=DataValStatusGen1_end([2:end],:);
csvwrite_with_headers(strcat('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\DataValGen1_',num2str(TempsVal),'s.csv'),[DataValGen1_end
DataValStatusGen1_end],LabelsGen1);
save('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\Gen1.mat','Gen1')
clearvars -except Sim_Sta_Inst FichierSimul NombreDeSimul DownS TempsApp TempsTest TempsVal
Prop* RandID*
```

```

%% Creation des fichiers de validation de test et d'apprenissage Gen 2

FichierSimul = dir('*.mat');
NombreDeSimul = length(FichierSimul);
DataAppGen2_end=zeros(1,72);
DataAppStatusGen2_end=0;
DataTestGen2_end=zeros(1,72);
DataTestStatusGen2_end=0;
DataValGen2_end=zeros(1,72);
DataValStatusGen2_end=0;

rand('twister', RandIDGen2);
VectAppInstab=sort(randsample(find(Sim_Sta_Inst(:,2)==1),PropApp(2,2)));
rand('twister', RandIDGen2);
VectAppSta=sort(randsample(find(Sim_Sta_Inst(:,2)==0),PropApp(1,2)));
Sim_Sta_Inst(VectAppSta,2)=999;
Sim_Sta_Inst(VectAppInstab,2)=999;
rand('twister', RandIDGen2);
VectTestInstab=sort(randsample(find(Sim_Sta_Inst(:,2)==1),PropTest(2,2)));
rand('twister', RandIDGen2);
VectTestSta=sort(randsample(find(Sim_Sta_Inst(:,2)==0),PropTest(1,2)));
Sim_Sta_Inst(VectTestSta,2)=999;
Sim_Sta_Inst(VectTestInstab,2)=999;
rand('twister', RandIDGen2);
VectValInstab=sort(randsample(find(Sim_Sta_Inst(:,2)==1),PropVal(2,2)));
rand('twister', RandIDGen2);
VectValSta=sort(randsample(find(Sim_Sta_Inst(:,2)==0),PropVal(1,2)));
Sim_Sta_Inst(VectValSta,2)=999;
Sim_Sta_Inst(VectValInstab,2)=999;

for k = 1:NombreDeSimul
    if(isempty(find(VectAppSta(:,1)==k))~=1||...
        isempty(find(VectAppInstab(:,1)==k))~=1)
        nom=FichierSimul(k,1).name;
        Gen2{k,1}=nom;
        Gen2{k,2}=length(DataAppStatusGen2_end);
        load(nom,'status','VectElectGen2');
        if(isempty(strfind(nom,'long'))==0)
            if length(VectElectGen2)>(4608)+3840*TempsApp
                DataAppGen2_end=[DataAppGen2_end;downsample(VectElectGen2...
                    ([4608:(4608)+3840*TempsApp],:),DownS)];
                DataAppStatusGen2_end=[DataAppStatusGen2_end;downsample(...
                    status([4608:(4608)+3840*TempsApp],2),DownS)];
            else
                DataAppGen2_end=[DataAppGen2_end;downsample(...
                    VectElectGen2([4608:end],:),DownS)];
                DataAppStatusGen2_end=[DataAppStatusGen2_end;downsample(...
                    status([4608:end],2),DownS)];
            end
        else
            if length(VectElectGen2)>(4224)+3840*TempsApp
                DataAppGen2_end=[DataAppGen2_end;downsample(VectElectGen2(...
                    [4224:(4224)+3840*TempsApp],:),DownS)];
                DataAppStatusGen2_end=[DataAppStatusGen2_end;downsample(...
                    status([4224:(4224)+3840*TempsApp],2),DownS)];
            else
                DataAppGen2_end=[DataAppGen2_end;downsample(VectElectGen2...
                    ([4224:end],:),DownS)];
                DataAppStatusGen2_end=[DataAppStatusGen2_end;downsample(...
                    status([4224:end],2),DownS)];
            end
        end
    elseif(isempty(find(VectTestSta(:,1)==k))~=1||...
        isempty(find(VectTestInstab(:,1)==k))~=1)

```

```

nom=FichierSimul(k,1).name;
Gen2{k,3}=nom;
Gen2{k,4}=length(DataTestStatusGen2_end);
load(nom,'status','VectElectGen2');
if isempty(strfind(nom, 'long'))==0
    if length(VectElectGen2)>(4608)+3840*TempsTest
        DataTestGen2_end=[DataTestGen2_end;downsample(...
            VectElectGen2([4608:(4608)+3840*TempsTest],:),DownS)];
        DataTestStatusGen2_end=[DataTestStatusGen2_end;downsample(...
            status([4608:(4608)+3840*TempsTest],2),DownS)];
    else
        DataTestGen2_end=[DataTestGen2_end;downsample(...
            VectElectGen2([4608:end],:),DownS)];
        DataTestStatusGen2_end=[DataTestStatusGen2_end;downsample(...
            status([4608:end],2),DownS)];
    end
else
    if length(VectElectGen2)>(4224)+3840*TempsTest
        DataTestGen2_end=[DataTestGen2_end;downsample(...
            VectElectGen2([4224:(4224)+3840*TempsTest],:),DownS)];
        DataTestStatusGen2_end=[DataTestStatusGen2_end;downsample(...
            status([4224:(4224)+3840*TempsTest],2),DownS)];
    else
        DataTestGen2_end=[DataTestGen2_end;downsample(...
            VectElectGen2([4224:end],:),DownS)];
        DataTestStatusGen2_end=[DataTestStatusGen2_end;...
            downsample(status([4224:end],2),DownS)];
    end
end
elseif isempty(find(VectValSta(:,1)==k)~=1)||...
    isempty(find(VectValInstab(:,1)==k)~=1)

    nom=FichierSimul(k,1).name;
    Gen2{k,5}=nom;
    Gen2{k,6}=length(DataValStatusGen2_end);
    load(nom,'status','VectElectGen2');
    if isempty(strfind(nom, 'long'))==0
        if length(VectElectGen2)>(4608)+3840*TempsVal
            DataValGen2_end=[DataValGen2_end;downsample(VectElectGen2(...
                [4608:(4608)+3840*TempsVal],:),DownS)];
            DataValStatusGen2_end=[DataValStatusGen2_end;downsample(...
                status([4608:(4608)+3840*TempsVal],2),DownS)];
        else
            DataValGen2_end=[DataValGen2_end;downsample(VectElectGen2(...
                [4608:end],:),DownS)];
            DataValStatusGen2_end=[DataValStatusGen2_end;downsample(...
                status([4608:end],2),DownS)];
        end
    else
        if length(VectElectGen2)>(4224)+3840*TempsVal
            DataValGen2_end=[DataValGen2_end;downsample(VectElectGen2(...
                [4224:(4224)+3840*TempsVal],:),DownS)];
            DataValStatusGen2_end=[DataValStatusGen2_end;downsample(...
                status([4224:(4224)+3840*TempsVal],2),DownS)];
        else
            DataValGen2_end=[DataValGen2_end;downsample(VectElectGen2(...
                [4224:end],:),DownS)];
            DataValStatusGen2_end=[DataValStatusGen2_end;downsample(...
                status([4224:end],2),DownS)];
        end
    end
end

```

```

        end
    end

end

DataAppGen2_end=DataAppGen2_end([2:end],:);
DataAppStatusGen2_end=DataAppStatusGen2_end([2:end],:);
load('rpsslackg1_1001_tripahse_lignea_30_long.mat','LabelsGen2');
LabelsGen2{1,73}='Status';
csvwrite_with_headers(strcat('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\DataAppGen2_',num2str(TempsApp),'s.csv'),[DataAppGen2_end
DataAppStatusGen2_end],LabelsGen2);
DataTestGen2_end=DataTestGen2_end([2:end],:);
DataTestStatusGen2_end=DataTestStatusGen2_end([2:end],:);
csvwrite_with_headers(strcat('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\DataTestGen2_',num2str(TempsTest),'s.csv'),[DataTestGen2_end
DataTestStatusGen2_end],LabelsGen2);
DataValGen2_end=DataValGen2_end([2:end],:);
DataValStatusGen2_end=DataValStatusGen2_end([2:end],:);
csvwrite_with_headers(strcat('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\DataValGen2_',num2str(TempsVal),'s.csv'),[DataValGen2_end
DataValStatusGen2_end],LabelsGen2);
save('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\Gen2.mat','Gen2')
clearvars -except Sim_Sta_Inst FichierSimul NombreDeSimul Downs TempsApp TempsTest TempsVal
Prop* RandID*

%% Creation des fichiers de validation de test et d'apprenissage Gen 3

FichierSimul = dir('*.mat');
NombreDeSimul = length(FichierSimul);
DataAppGen3_end=zeros(1,72);
DataAppStatusGen3_end=0;
DataTestGen3_end=zeros(1,72);
DataTestStatusGen3_end=0;
DataValGen3_end=zeros(1,72);
DataValStatusGen3_end=0;

rand('twister', RandIDGen3);
VectAppInstab=sort(randsample(find(Sim_Sta_Inst(:,3)==1),PropApp(2,3)));
rand('twister', RandIDGen3);
VectAppSta=sort(randsample(find(Sim_Sta_Inst(:,3)==0),PropApp(1,3)));
Sim_Sta_Inst(VectAppSta,3)=999;
Sim_Sta_Inst(VectAppInstab,3)=999;
rand('twister', RandIDGen3);
VectTestInstab=sort(randsample(find(Sim_Sta_Inst(:,3)==1),PropTest(2,3)));
rand('twister', RandIDGen3);
VectTestSta=sort(randsample(find(Sim_Sta_Inst(:,3)==0),PropTest(1,3)));
Sim_Sta_Inst(VectTestSta,3)=999;
Sim_Sta_Inst(VectTestInstab,3)=999;
rand('twister', RandIDGen3);
VectValInstab=sort(randsample(find(Sim_Sta_Inst(:,3)==1),PropVal(2,3)));
rand('twister', RandIDGen3);
VectValSta=sort(randsample(find(Sim_Sta_Inst(:,3)==0),PropVal(1,3)));
Sim_Sta_Inst(VectValSta,3)=999;
Sim_Sta_Inst(VectValInstab,3)=999;

for k = 1:NombreDeSimul
    if (isempty(find(VectAppSta(:,1)==k))~=1 || ...
        isempty(find(VectAppInstab(:,1)==k))~=1)
        nom=FichierSimul(k,1).name;
        Gen3{k,1}=nom;
    end
end

```

```

Gen3{k,2}=length(DataAppStatusGen3_end);
load(nom,'status','VectElectGen3');
if isempty(strfind(nom, 'long')==0)
    if length(VectElectGen3)>(4608)+3840*TempsApp
        DataAppGen3_end=[DataAppGen3_end;downsample(...
            VectElectGen3([4608:(4608)+3840*TempsApp],:),DownS)];
        DataAppStatusGen3_end=[DataAppStatusGen3_end;downsample(...
            status([4608:(4608)+3840*TempsApp],3),DownS)];
    else
        DataAppGen3_end=[DataAppGen3_end;downsample(VectElectGen3(...
            [4608:end],:),DownS)];
        DataAppStatusGen3_end=[DataAppStatusGen3_end;downsample(...
            status([4608:end],3),DownS)];
    end
else
    if length(VectElectGen3)>(4224)+3840*TempsApp

        DataAppGen3_end=[DataAppGen3_end;downsample(VectElectGen3(...
            [4224:(4224)+3840*TempsApp],:),DownS)];
        DataAppStatusGen3_end=[DataAppStatusGen3_end;downsample(...
            status([4224:(4224)+3840*TempsApp],3),DownS)];

    else
        DataAppGen3_end=[DataAppGen3_end;downsample(VectElectGen3(...
            [4224:end],:),DownS)];
        DataAppStatusGen3_end=[DataAppStatusGen3_end;downsample(...
            status([4224:end],3),DownS)];
    end
end
elseif(isempty(find(VectTestSta(:,1)==k))~=1||...
    isempty(find(VectTestInstab(:,1)==k))~=1)
    nom=FichierSimul(k,1).name;
    Gen3{k,3}=nom;
    Gen3{k,4}=length(DataTestStatusGen3_end);
    load(nom,'status','VectElectGen3');
    if isempty(strfind(nom, 'long')==0)
        if length(VectElectGen3)>(4608)+3840*TempsTest
            DataTestGen3_end=[DataTestGen3_end;downsample(...
                VectElectGen3([4608:(4608)+3840*TempsTest],:),DownS)];
            DataTestStatusGen3_end=[DataTestStatusGen3_end;downsample(...
                status([4608:(4608)+3840*TempsTest],3),DownS)];
        else
            DataTestGen3_end=[DataTestGen3_end;downsample(...
                VectElectGen3([4608:end],:),DownS)];
            DataTestStatusGen3_end=[DataTestStatusGen3_end;downsample(...
                status([4608:end],3),DownS)];
        end
    else
        if length(VectElectGen3)>(4224)+3840*TempsTest

            DataTestGen3_end=[DataTestGen3_end;downsample(...
                VectElectGen3([4224:(4224)+3840*TempsTest],:),DownS)];
            DataTestStatusGen3_end=[DataTestStatusGen3_end;downsample(...
                status([4224:(4224)+3840*TempsTest],3),DownS)];

        else
            DataTestGen3_end=[DataTestGen3_end;downsample(...
                VectElectGen3([4224:end],:),DownS)];
            DataTestStatusGen3_end=[DataTestStatusGen3_end;downsample(...
                status([4224:end],3),DownS)];
        end
    end
end
elseif(isempty(find(VectValSta(:,1)==k))~=1||...
    isempty(find(VectValInstab(:,1)==k))~=1)

```

```

    nom=FichierSimul(k,1).name;
    Gen3{k,5}=nom;
    Gen3{k,6}=length(DataValStatusGen3_end);
    load(nom,'status','VectElectGen3');
    if isempty(strfind(nom,'long')==0)
        if length(VectElectGen3)>(4608)+3840*TempsVal
            DataValGen3_end=[DataValGen3_end;downsample(VectElectGen3(...
                [4608:(4608)+3840*TempsVal],:),DownS)];
            DataValStatusGen3_end=[DataValStatusGen3_end;downsample(...
                status([4608:(4608)+3840*TempsVal],3),DownS)];
        else
            DataValGen3_end=[DataValGen3_end;downsample(VectElectGen3(...
                [4608:end],:),DownS)];
            DataValStatusGen3_end=[DataValStatusGen3_end;downsample(...
                status([4608:end],3),DownS)];
        end
    else
        if length(VectElectGen3)>(4224)+3840*TempsVal
            DataValGen3_end=[DataValGen3_end;downsample(VectElectGen3(...
                [4224:(4224)+3840*TempsVal],:),DownS)];
            DataValStatusGen3_end=[DataValStatusGen3_end;downsample(...
                status([4224:(4224)+3840*TempsVal],3),DownS)];
        else
            DataValGen3_end=[DataValGen3_end;downsample(VectElectGen3(...
                [4224:end],:),DownS)];
            DataValStatusGen3_end=[DataValStatusGen3_end;downsample(...
                status([4224:end],3),DownS)];
        end
    end
end

end

DataAppGen3_end=DataAppGen3_end([2:end],:);
DataAppStatusGen3_end=DataAppStatusGen3_end([2:end],:);
load('rpsslackg1_1001_tripahse_ligne30_long.mat','LabelsGen3');
LabelsGen3{1,73}='Status';
csvwrite_with_headers(strcat('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\DataAppGen3_',num2str(TempsApp),'s.csv'),[DataAppGen3_end
DataAppStatusGen3_end],LabelsGen3);
DataTestGen3_end=DataTestGen3_end([2:end],:);
DataTestStatusGen3_end=DataTestStatusGen3_end([2:end],:);
csvwrite_with_headers(strcat('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\DataTestGen3_',num2str(TempsTest),'s.csv'),[DataTestGen3_end
DataTestStatusGen3_end],LabelsGen3);
DataValGen3_end=DataValGen3_end([2:end],:);
DataValStatusGen3_end=DataValStatusGen3_end([2:end],:);
csvwrite_with_headers(strcat('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\DataValGen3_',num2str(TempsVal),'s.csv'),[DataValGen3_end
DataValStatusGen3_end],LabelsGen3);
save('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\Gen3.mat','Gen3')
clearvars -except Sim_Inst FichierSimul NombreDeSimul Downs TempsApp TempsTest TempsVal
Prop* RandID*

%% Creation des fichiers de validation de test et d'apprenissage Gen 4

FichierSimul = dir('*.mat');
NombreDeSimul = length(FichierSimul);
DataAppGen4_end=zeros(1,72);

```

```

DataAppStatusGen4_end=0;
DataTestGen4_end=zeros(1,72);
DataTestStatusGen4_end=0;
DataValGen4_end=zeros(1,72);
DataValStatusGen4_end=0;

rand('twister', RandIDGen4);
VectAppInstab=sort(randsample(find(Sim_Sta_Inst(:,4)==1),PropApp(2,4)));
rand('twister', RandIDGen4);
VectAppSta=sort(randsample(find(Sim_Sta_Inst(:,4)==0),PropApp(1,4)));
Sim_Sta_Inst(VectAppSta,4)=999;
Sim_Sta_Inst(VectAppInstab,4)=999;
rand('twister', RandIDGen4);
VectTestInstab=sort(randsample(find(Sim_Sta_Inst(:,4)==1),PropTest(2,4)));
rand('twister', RandIDGen4);
VectTestSta=sort(randsample(find(Sim_Sta_Inst(:,4)==0),PropTest(1,4)));
Sim_Sta_Inst(VectTestSta,4)=999;
Sim_Sta_Inst(VectTestInstab,4)=999;
rand('twister', RandIDGen4);
VectValInstab=sort(randsample(find(Sim_Sta_Inst(:,4)==1),PropVal(2,4)));
rand('twister', RandIDGen4);
VectValSta=sort(randsample(find(Sim_Sta_Inst(:,4)==0),PropVal(1,4)));
Sim_Sta_Inst(VectValSta,4)=999;
Sim_Sta_Inst(VectValInstab,4)=999;

for k = 1:NombreDeSimul
    if isempty(find(VectAppSta(:,1)==k))~=1 || ...
        isempty(find(VectAppInstab(:,1)==k))~=1
        nom=FichierSimul(k,1).name;
        Gen4{k,1}=nom;
        Gen4{k,2}=length(DataAppStatusGen4_end);
        load(nom,'status','VectElectGen4');
        if isempty(strfind(nom,'long')==0)
            if length(VectElectGen4)>(4608)+3840*TempsApp
                DataAppGen4_end=[DataAppGen4_end;downsample(VectElectGen4(...
                    [4608:(4608)+3840*TempsApp],:),DownS)];
                DataAppStatusGen4_end=[DataAppStatusGen4_end;downsample(...
                    status([4608:(4608)+3840*TempsApp],4),DownS)];
            else
                DataAppGen4_end=[DataAppGen4_end;downsample(VectElectGen4(...
                    [4608:end],:),DownS)];
                DataAppStatusGen4_end=[DataAppStatusGen4_end;downsample(...
                    status([4608:end],4),DownS)];
            end
        else
            if length(VectElectGen4)>(4224)+3840*TempsApp
                DataAppGen4_end=[DataAppGen4_end;downsample(VectElectGen4(...
                    [4224:(4224)+3840*TempsApp],:),DownS)];
                DataAppStatusGen4_end=[DataAppStatusGen4_end;downsample(...
                    status([4224:(4224)+3840*TempsApp],4),DownS)];
            else
                DataAppGen4_end=[DataAppGen4_end;downsample(VectElectGen4(...
                    [4224:end],:),DownS)];
                DataAppStatusGen4_end=[DataAppStatusGen4_end;downsample(...
                    status([4224:end],4),DownS)];
            end
        end
    elseif isempty(find(VectTestSta(:,1)==k))~=1 || ...
        isempty(find(VectTestInstab(:,1)==k))~=1
        nom=FichierSimul(k,1).name;
        Gen4{k,3}=nom;
        Gen4{k,4}=length(DataTestStatusGen4_end);
        load(nom,'status','VectElectGen4');

```

```

if(isempty(strfind(nom, 'long'))==0)
    if length(VectElectGen4)>(4608)+3840*TempsTest
        DataTestGen4_end=[DataTestGen4_end;downsample(...
            VectElectGen4([4608:(4608)+3840*TempsTest],:),DownS)];
        DataTestStatusGen4_end=[DataTestStatusGen4_end;downsample(...
            status([4608:(4608)+3840*TempsTest],4),DownS)];
    else
        DataTestGen4_end=[DataTestGen4_end;downsample(...
            VectElectGen4([4608:end],:),DownS)];
        DataTestStatusGen4_end=[DataTestStatusGen4_end;downsample(...
            status([4608:end],4),DownS)];
    end
else
    if length(VectElectGen4)>(4224)+3840*TempsTest
        DataTestGen4_end=[DataTestGen4_end;downsample(...
            VectElectGen4([4224:(4224)+3840*TempsTest],:),DownS)];
        DataTestStatusGen4_end=[DataTestStatusGen4_end;downsample(...
            status([4224:(4224)+3840*TempsTest],4),DownS)];
    else
        DataTestGen4_end=[DataTestGen4_end;downsample(...
            VectElectGen4([4224:end],:),DownS)];
        DataTestStatusGen4_end=[DataTestStatusGen4_end;downsample(...
            status([4224:end],4),DownS)];
    end
end

elseif(isempty(find(VectValSta(:,1)==k))~=1||...
    isempty(find(VectValInstab(:,1)==k))~=1)

    nom=FichierSimul(k,1).name;
    Gen4{k,5}=nom;
    Gen4{k,6}=length(DataValStatusGen4_end);
    load(nom,'status','VectElectGen4');
    if(isempty(strfind(nom, 'long'))==0)
        if length(VectElectGen4)>(4608)+3840*TempsVal
            DataValGen4_end=[DataValGen4_end;downsample(...
                VectElectGen4([4608:(4608)+3840*TempsVal],:),DownS)];
            DataValStatusGen4_end=[DataValStatusGen4_end;downsample(...
                status([4608:(4608)+3840*TempsVal],4),DownS)];
        else
            DataValGen4_end=[DataValGen4_end;downsample(...
                VectElectGen4([4608:end],:),DownS)];
            DataValStatusGen4_end=[DataValStatusGen4_end;downsample(...
                status([4608:end],4),DownS)];
        end
    else
        if length(VectElectGen4)>(4224)+3840*TempsVal
            DataValGen4_end=[DataValGen4_end;downsample(VectElectGen4(...
                [4224:(4224)+3840*TempsVal],:),DownS)];
            DataValStatusGen4_end=[DataValStatusGen4_end;downsample(...
                status([4224:(4224)+3840*TempsVal],4),DownS)];
        else
            DataValGen4_end=[DataValGen4_end;downsample(VectElectGen4(...
                [4224:end],:),DownS)];
            DataValStatusGen4_end=[DataValStatusGen4_end;downsample(...
                status([4224:end],4),DownS)];
        end
    end
end
end
end
end

```

```

DataAppGen4_end=DataAppGen4_end([2:end],:);
DataAppStatusGen4_end=DataAppStatusGen4_end([2:end],:);
load('rpsslackg1_1001_tripahse_ligne_30_long.mat','LabelsGen4');
LabelsGen4{1,73}='Status';
csvwrite_with_headers(strcat('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\DataAppGen4_',num2str(TempsApp),'s.csv'),[DataAppGen4_end
DataAppStatusGen4_end],LabelsGen4);
DataTestGen4_end=DataTestGen4_end([2:end],:);
DataTestStatusGen4_end=DataTestStatusGen4_end([2:end],:);
csvwrite_with_headers(strcat('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\DataTestGen4_',num2str(TempsTest),'s.csv'),[DataTestGen4_end
DataTestStatusGen4_end],LabelsGen4);
DataValGen4_end=DataValGen4_end([2:end],:);
DataValStatusGen4_end=DataValStatusGen4_end([2:end],:);
csvwrite_with_headers(strcat('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\DataValGen4_',num2str(TempsVal),'s.csv'),[DataValGen4_end
DataValStatusGen4_end],LabelsGen4);
save('C:\Documents and Settings\Administrateur\Bureau\Donnes
RPS\Donnes\CSV\Gen4.mat','Gen4')
clearvars -except Sim_Sta_Inst FichierSimul NombreDeSimul Downs TempsApp TempsTest TempsVal
Prop* RandID*

```